TECHNICAL UNIVERSITY OF KAISERSLAUTERN

MASTER THESIS

Dynamic risk assessment at intersections

Author:

Siva Kalyani KONETI

Supervisor:

Prof. Peter LIGGESMEYER

Patrik FETH

A thesis submitted in partial fulfillment of the requirements for the degree of Master in the Commercial Vehicle Technology

in the

Fraunhofer Institute for Experimental Software Engineering

Software Engineering Dependability

November 9, 2018

Declaration of Authorship

I, Siva Kalyani KONETI, declare that this thesis titled, “Dynamic risk assessment at intersections” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"In a day, when you don't come across any problems - you can be sure that you are traveling in a wrong path"

Swami Vivekananda

Abstract

Autonomous driving has a great potential for reducing traffic accidents. This paves a way for the improvement of road safety. However, the very notion of risk is not always a clearly defined concept. Existing work have shown how risk metrics used for performing the risk assessments are domain specific and are defined on a vehicular level. On a global level for an intersection with multiple vehicles, these risk metrics cannot be applied without further modifications.

First part of the work involves creating a test scenario using SUMO simulator, importing this scenario into the Risk Metric Calculator (RMC), deciding on the prediction models that will be used for carrying out the risk assessment and defining a new mission level risk metric to carry out a Dynamic Risk Assessment at the considered intersection. To this end, intended trajectory prediction model and reachable area prediction model are used for risk assessment and a new risk metric MTTC (Mission Level Time to Collision) is defined.

The concept of Conditional Safety Certificates (ConSerts) is applied in the second half of the work. The result of evaluation of these ConSerts at runtime is used to determine if the Dynamic Risk Assessment (DRA) on different confidence levels at the intersection could be performed or not. To achieve this a communication between the SUMO simulator and ConSerts evaluation and visualization services has been created.

Acknowledgements

I am using this opportunity to express my gratitude to everyone who supported me to finish this work.

First and above all, I would like to thank the God Almighty to whom I owe my very existence for providing me the strength and this opportunity to undertake this research study and to persevere and complete it satisfactorily.

I would like to express my gratitude towards Prof. Dr.Ing. habil. Peter Liggesmeyer. I am grateful for being able to work on an interesting topic in his group.

I am grateful to Frauhofer IESE, for allowing me the opportunity to conduct this work at Fraunhofer, utlilising their resources.

My sincere thanks also to my supervisor Patrik Feth, for his patience, motivation, encouraging guidance and consistent support throughout.

Last but not the least, I would also like to thank my family and friends who all greatly supported me during my times I needed encouragement and boost of confidence to complete my studies and life in general.

Contents

List of Figures

List of Abbreviations

GPS	**G**lobal **P**ositioning **S**ystem
ITS	**I**ntelligent **T**ransport **S**ystems
ICT	**I**nformation and **C**ommunication **T**echnologies
C-ITS	**C**ooperative **I**ntelligent **T**ransport **S**ystems
CAS	**C**ollision **A**voidance **S**ystems
DGPS	**D**ifferential **G**lobal **P**ositioning **S**ystems
V2V	**V**ehicle to **V**ehicle
V2I	**V**ehicle to **I**nfrastructure
SUMO	**S**imulation **o**f **U**rban **M**obility
NETEDIT	**NET**work **EDIT**or
XML	e**X**tensible **M**arkup **L**anguage
TTC	**T**ime **T**o **C**ollision
CA	**C**ollision **A**voidance
TTX	**T**ime **T**o **X**
ADAS	**A**dvanced **D**river **A**ssistance **S**ystems
TTR	**T**ime **T**o **R**eact
TTB	**T**ime **T**o **B**rake
TTK	**T**ime **T**o **K**ickdown
WTTC	**W**orst **T**ime **T**o **C**ollision
TTCCP	**T**ime **T**o **C**ritical **C**ollision **P**robability
PCE	**P**otential **C**ollision **E**nergy
TET	**T**ime **E**xposed **T**ime To Collision
TIT	**T**ime **I**integrated **T**ime To Collision
ETTC	**E**nhanced **T**ime **T**o **C**ollision
DH	**D**istance **H**eadway
TEDH	**T**ime **E**xposed **D**istance **H**eadway

TIDH	**T**ime **I**integrated **D**istance **H**eadway
TES	**T**ime **E**xposed **S**peed
DRA	**D**ynamic **R**isk **A**ssessment
SWM	**S**afety **W**orld **M**odel
ITTC	**I**ntersection **T**ime **T**o **C**ollision
LTTC	**L**ocal **T**ime **T**o **C**ollision
GTTC	**G**eneral **T**ime **T**o **C**ollision
MTTC	**M**ission level **T**ime **T**o **C**ollision
VY	**V**ehicle **Y**aw
VT	**V**ehicle **T**rajectory
VS	**V**ehicle **S**peed
VP	**V**ehicle **P**osition
SC	**S**tereo **C**amera
RMC	**R**isk **M**etric **C**alculator
HC	**H**igh **C**onfidence
MC	**M**id **C**onfidence
LC	**L**ow **C**onfidence
GUI	**G**rafic **U**ser **I**nterface
EW	**E**ast to **W**est
NS	**N**orth to **S**outh
WE	**W**est to **E**ast
SN	**S**outh to **N**orth
EWTR	**E**ast **W**est **T**urn **R**ight
VS HC	**V**ehicle **S**peed **H**igh **C**onfidence
VP HC	**V**ehicle **P**osition **H**igh **C**onfidence
VT HC	**V**ehicle **T**rajectory **H**igh **C**onfidence
VY HC	**V**ehicle **Y**aw **H**igh **C**onfidence
DRA HC	**D**ynamic **R**isk **A**ssessment **H**igh **C**onfidence
VS MC	**V**ehicle **S**peed **M**id **C**onfidence
VP MC	**V**ehicle **P**osition **M**id **C**onfidence
VT MC	**V**ehicle **T**rajectory **M**id **C**onfidence

VY MC	**V**ehicle **Y**aw **M**id **C**onfidence
DRA MC	**D**ynamic **R**isk **A**ssessment **M**id **C**onfidence
VS LC	**V**ehicle **S**peed **L**ow **C**onfidence
VP LC	**V**ehicle **P**osition **L**ow **C**onfidence
VT LC	**V**ehicle **T**rajectory **L**ow **C**onfidence
VY LC	**V**ehicle **Y**aw **L**ow **C**onfidence
DRA LC	**D**ynamic **R**isk **A**ssessment **L**ow **C**onfidence
Infra	**Infra**structure

Dedicated to my beloved parents, Aruna Koneti and Ravi Kumar Koneti.....

Chapter 1

Introduction

This chapter gives a brief description about the concepts of cooperative intersection management and collision avoidance for autonomous vehicles at the intersection. Section 1.4 explains the need for performing a Dynamic Risk Assessment (DRA) at an intersection. Section 1.5 gives a short input on the simulator that is being used in the current work to carry out the simulations based on the decided scenarios.

1.1 Autonomous vehicles and non signalized intersections

An autonomous car also known as a driver less car, robot car, self-driving car or simply an autonomous vehicle is a vehicle that can guide itself without human conduction. This kind of vehicle is slowly becoming an actual reality paving a way for development of new systems, where computers take over the art of driving using various kinds of technologies. In order to achieve it, they need a great variety of sensors to help with navigation and may use these sensors and other equipment to avoid collisions. Driver less car designers are faced with a challenge to produce control systems capable of analyzing sensory data to provide accurate detection of other vehicles and the road ahead or whatever information is needed to plan a driving behavior.

Safety at non signalized intersections is of critical importance. An intersection is a junction where two or more roads either meet or cross. A non signalized intersection is an intersection that is not managed by traffic light system or any other stop signs. Traffic intersections are complicated because different flows of traffic compete for the same space (*Transportation Engineering I* 2009). However, with autonomous vehicles which are equipped with sensors that help with the communication with other vehicles and traffic management systems at an intersection, it could be easier to anticipate when each vehicle would reach the crossing and route them efficiently.

1.2 Cooperative intersection management

Intersection management is one of the most challenging problems within the transport system for keeping traffic safety and smoothing traffic flow. Methods based on traffic lights have been efficient but are not able to deal with the growing mobility and social challenges. Recent advancements in information and communication technologies (ICT) have enabled new methods, such as the prevalence of intelligent transport systems (ITS) for intersection management. With standardization, research and demonstration efforts on Cooperative - intelligent transport systems (C–ITS) together with the advancements in vehicle control technologies, intersection safety and efficiency can be improved significantly. Simulating the traffic, facilitates the evaluation of changes in infrastructure as well as policy changes before implementing them on the road (Chen and Englund, 2016). For example, the environmental zones effectiveness or algorithms used for traffic light control can be tested and optimized in a simulation before being deployed in the real world (Krajzewicz et al., 2012).

The advancements of automation and communications have enabled cooperative intersection management, where road users, infrastructure, and traffic control centers are able to communicate and coordinate the traffic safely and efficiently (Chen and Englund, 2016). Methods such as time slots and space reservation, trajectory planning, virtual traffic lights, vehicle collision warning and avoidance methods are various techniques for cooperative intersection management. A cooperative vehicle intersection control scheme without using any traffic lights, based on model predictive control theory presented in (Zheng et al., 2017) assumes that all vehicles are fully automated and that a control unit is installed at the intersection to coordinate the vehicle movement which is a little similar to the architecture considered for the thesis work.

1.3 Intersection collision avoidance

Intersection Collision Avoidance Systems (CAS) address hazardous situations occurring in the vicinity of intersections. This user service warns a driver of imminent collisions when approaching or crossing intersections that have some form of traffic control. Intersection CAS would provide the driver with assistance in avoiding collisions at intersections due to inattention, faulty perception, obstructed views or intoxication (Li, 2000).

A reliable collision detection and avoidance system is indispensable for intelligent vehicles. Two main requirements should be satisfied, in order to ensure that an intelligent vehicle performs in a collision-free and smooth manner. The local environment, including the road structure and drivable areas, should be perceived and understood first, which is helpful in the decision-making process. Secondly, the dynamic obstacles must be reliably detected and predicted using information from on-board sensors. These sensors allow for potential collisions to be detected and avoided in a timely manner (Huang et al., 2017).

High-percentage crashes, such as road departure and rear end, have radar and camera-based forward collision systems available (Hafner et al., 2013). Also, there are fewer new technologies addressing side-impact collisions at intersections . Various technologies such as Differential global positioning systems (DGPS), roadside sensors, etc, are being investigated for application to collision avoidance. It becomes necessarily important that the vehicles exchange dynamic information such as speed, acceleration, position and direction in real time. To this end, the goal being able to avoid collisions, using wireless communication technologies the vehicles can inform each other about how far they are from the intersection and receive the information from the signal lights dynamically. The vehicles thus, can also inform each other about the status of the intersection as wireless communications do not require a line-of-sight.

Particularly in case of autonomous vehicles, Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications are setting the basis for establishing technology. This is achieved by having vehicles cooperate with each other and with the surrounding infrastructure, sharing information about the environment and improving overall situational awareness. Therefore, intelligent transportation systems for inter vehicle cooperative (active) safety have been the subject of intense research world-wide in government and industry (Hafner et al., 2013).

The two main strategies for active safety systems could be categorised as telematics and sensing. In telematics, neighbourhood vehicles and infrastructure send information about their position, velocity or status (red light signal, turn status, etc.). This type of communication is described as V2X(Vehicle to Vehicle, Vehicle to Infrastructure) communication. The advantage of this communication method is that the information is accurate. But a major disadvantage is that not all the objects on the road are equipped with the necessary hardware. Also there are other issues like communication security, data integrity that make them vulnerable to threats. The other approach is to equip the subject vehicle with sensors like radar, camera, laser scanner, for perception of the environment. These systems do not have to rely on other vehicles that are in adjacent or neighbouring them. But the major challenge in this approach is that these systems must deal with extracting reliable information, even during bad weather conditions like fog, sleet, rain, etc,.

1.4 Need for dynamic risk assessment at the intersection

Autonomous driving technologies have a great potential for reducing traffic accidents and, thus, improve road safety (Huang et al., 2017). However, the very notion of risk is not always a clearly defined concept. Risk in general terms could be defined as something that is undesirable (for example an action), which could lead to a loss in it's own way. Safety on the other hand is to be able to stay protected from any undesirable events. It can hence be said that lower the risk, more safe is a system. The assessment of risk of a situation takes place in the present but is always considering a potential unwanted event in the future. Even though few steps can be taken to reduce and eliminate hazards, at development time there are still some risks that are unpredictable and are difficult to control. This gives the motivation to perform a dynamic risk assessment.

It is generally accepted that it cannot be completely guaranteed that a safety critical technical system won't cause harm to itself or it's environment. There is always residual risk associated. In any of such situations, the ability to carry out a dynamic risk assessment allows to identify a potentially dangerous environment or situation in order to take appropriate steps to leave the environment or to remove the risk before it causes any accident or incident. In the context of intelligent vehicles, it is generally associated with the idea that a situation may be dangerous for the driver, i.e. may result in harm or injury which are caused mostly by collisions with the other vehicles in the current situation or environment (Feth, Schneider, and Adler, 2017).

1.5 Simulator used for realization - SUMO

"Simulation of Urban MObility" or "SUMO" for short, is an open source, microscopic and multi-modal traffic simulation. It allows to simulate how a given traffic demand which consists of single vehicles moves through a given road network. The simulation allows addressing a large set of traffic management topics. It is purely microscopic: each vehicle is modelled explicitly, has an own route and moves individually through the network. Simulations are deterministic by default but there are various options for introducing randomness (Krajzewicz et al., 2012). It is mainly developed by employees of the Institute of Transportation Systems at the German Aerospace Centre.

Traffic and intersection behavior can be defined and also various scenarios can be developed using NETEDIT, a visual network editor vs. XML, to build the lanes, junctions, to place detectors, etc in SUMO simulator. Also there is an useful library called TraCI to work with vehicle related parameters provided by SUMO simulator which is used in the current work. TraCI is a short term for "Traffic Control Interface". It gives access to a running road traffic simulation, allowing to retrieve values of simulated objects and also to manipulate their behaviour. By using TraCI, we can define our own rules for the vehicles and for the behaviour of the intersection in SUMO simulator.

Chapter 2

Related Work

As discussed in Chapter 1, collision avoidance is the aim for autonomous vehicles with regards to safety. The approach to risk assessment based on collision prediction according to (Lefèvre, Vasquez, and Laugier, 2014) are composed of two steps:

- Predicting the potential future trajectories for all the moving entities in the scene.
- Detect collision between each pair of trajectories and derive a risk estimate based on the overall chance of collision.

Thus the choice of a risk assessment method is tightly coupled with the choice of a motion model.

2.1 Motion models for collision prediction

Motor vehicle travel is the primary means of transportation, providing an unparalleled degree of mobility. Yet for all its advantages, motor vehicle crashes are still the leading cause of death among individuals. Numbers in (Transport, 2004) suggests that majority of the crashes occur due to poor judgement and speeding.

Active safety systems can help drivers in preventing such collisions. Autonomous driving technologies have a great potential for reducing traffic accidents and, thus, improve road safety. Any action performed by a CA system will be called an intervention. Depending on application and the type of intervention considered, the metric for measuring collision threat and the decision making algorithm might vary significantly (Jansson, 2005).

A closely related concept is that of risk, which can be intuitively understood as the likelihood and severity of the damage that a vehicle of interest may suffer in the future. From this definition it is clear that, in order to assess the risk associated with a particular situation, it is necessary to have mathematical models which allow us to predict how this situation will evolve in the future (Lefèvre, Vasquez, and Laugier, 2014).

The proposed classification in (Lefèvre, Vasquez, and Laugier, 2014) is based on the semantics used to define motion and risk. The tradeoff between model completeness and real-time constraints, and the fact that the choice of a risk assessment method is influenced by the selected motion model is also pointed out. The three motion based models are explained below with reference to the figure 2.1 :

- Physics-based motion models: are the simplest models, they consider that the motion of vehicles only depends on the laws of physics. They assume a constant speed and orientation for the cars (Lefèvre, Vasquez, and Laugier, 2014).
- Maneuver-based motion models: are more advanced as they consider that the future motion of a vehicle also depends on the maneuver that the driver intends to perform. This type of model assumes that the black car goes straight and the blue car turns left (Lefèvre, Vasquez, and Laugier, 2014).
- Interaction-aware motion models: take into account the inter-dependencies between vehicles maneuvers. It assumes that the black car goes straight, that the blue car turns left and that the joint motion of the cars is constrained by the traffic rules (Lefèvre, Vasquez, and Laugier, 2014).

2.1.1 Physics based motion models for trajectory prediction

Future motion in case of physics based motion models is predicted using dynamic and kinetic models linking some control inputs (e.g. steering, acceleration), car properties (e.g. weight) and external conditions (e.g. friction coefficient of the road surface) to the evolution of the state of the vehicle (e.g. position, heading, speed). These are the most commonly used motion models for trajectory prediction and collision risk estimation in the context of road safety (Lefèvre, Vasquez, and Laugier, 2014).

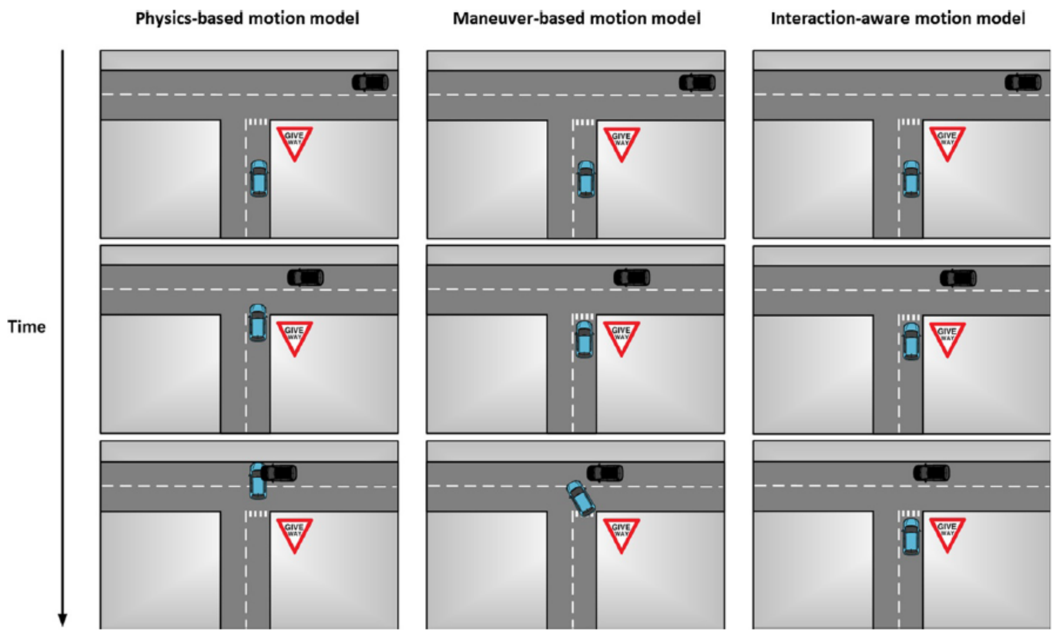

FIGURE 2.1: Examples of motion prediction with the different types of motion models (Lefèvre, Vasquez, and Laugier, 2014)

The evolution models (dynamic models and kinematic models) are used for trajectory prediction. Trajectory prediction is of interest to us as we will use this concept in the current work.

In general, trajectory is represented as a sequence of states visited by the vehicle, parameterised by time and, possibly, velocity. Trajectory planning (also known as trajectory generation) is concerned with the real-time planning of the actual vehicle's transition from one feasible state to the next, satisfying the vehicle's kinematic limits based on vehicle dynamics and constrained by the navigation comfort, to lane boundaries and traffic rules, while avoiding, at the same time, obstacles including other road users as well as ground roughness and ditches. Trajectory planning is parameterised by time as well as acceleration or velocity and is frequently referred to as motion planning (Katrakazas et al., 2015).

(Lefèvre, Vasquez, and Laugier, 2014) suggests that a straightforward manner to predict the future trajectory of a vehicle is to apply an evolution model to the current state of the vehicle, assuming that the current state is perfectly known and that the evolution model is a perfect representation of the motion of the vehicle. Advantage being the computational efficiency, the predicted trajectories however are not reliable for long term prediction (more than one second) . This method is close to the method used in this proposed work. Further is discussed in chapter 4.

Uncertainity on the current vehicle state and its evolution could be modeled by a normal distribution in case of Gaussian noise simulation. In the general cases when no assumptions are made on the linearity of the models or on the Gaussianity of the uncertainties, the analytical expression for the distribution on the predicted states is usually not known. Monte Carlo methods provide tools to approximate this distribution. The main limitation is that all the above models are limited to short term motion prediction (Lefèvre, Vasquez, and Laugier, 2014).

2.1.2 Maneuver based motion models

Manoeuvre is a high-level characterisation of the motion of the vehicle with regards to the position and speed of the vehicle on the road. Examples of manoeuvres include 'going straight', 'turning', 'overtaking' etc. A manoeuvre is considered nominal (acceptable in this case) if it is performed safely according to traffic or other rules. Hence, manoeuvre planning addresses the problem of taking the best high-level decision for the vehicle, while taking into account the path that is specified from path planning (Katrakazas et al., 2015).

Maneuver-based motion models represent vehicles as independent maneuvering entities. They assume that the vehicular motion on the road network corresponds to a series of maneuvers executed independently from the other vehicles. Hence a maneuver can be defined as "a physical movement or series of moves requiring skill and care" according to (Schreier, 2016) .

Trajectory prediction with maneuver based motion models is based on the early recognition of the maneuvers which the drivers intend to perform at the intersection. (Lefèvre, Vasquez, and Laugier, 2014) explains the concept of prototype trajectories. The idea of prototype trajectories is that on a road network, the trajectories of vehicles can be grouped into a finite set of clusters. Each cluster corresponding to a typical motion pattern which are represented using prototype trajectories that are learned from data during a training phase. Subsequently prediction can be performed online if a partial trajectory is given by finding the most likely motion pattern(s) and using the prototype trajectories as a model for future motion.

However one assumption that causes discrepency is that vehicles move independently from each other. This does not hold as the vehicles share the road with other vehicles and the maneuvers performed by one vehicle will compulsorily and necessarily influence the maneuvers of other vehicles.

2.1.3 Interaction aware motion models

Interaction-aware motion models represent vehicles as maneuvering entities which interact with each other. The subject vehicular motion is assumed to be influenced by the motion of the other vehicles in the scene. Considering the dependencies between the vehicles helps to come up with a better interpretation of their motion compared to the maneuver-based motion models. As a result, a better understanding of the situation and a more reliable evaluation of the risk is possible (Lefèvre, Vasquez, and Laugier, 2014).

2.2 Risk assessment techniques based on collision prediction for intelligent vehicles

The analytical solution at a specific time for the state of the vehicles can be easily derived by solving the linear differential equations of the motion model in case of binary collision prediction. However, the equations of motion are too complex. A common approach is to discretize the trajectories and iteratively check for a collision at each discrete timestep. Collisions thus can be detected in a simple manner by defining a threshold on the distance between two points (from two trajectories at the same timestep) (Lefèvre, Vasquez, and Laugier, 2014).

Considering the uncertainty on the future motion of vehicles, the collision risk can be computed in a probabilistic manner. In the case of stochastic reachable sets, the collision probability can be computed on a discretized position space by calculating the probability that the center of both vehicles is in the same cell, for all the possible combinations of cells. Whereas with the geometric version of reachable sets, the collision probability can be measured as the percentage of overlap between the geometric shapes representing the future motion of vehicles (Lefèvre, Vasquez, and Laugier, 2014).

Furthermore, based on the final application one can compute the risk of colliding with a specific vehicle or sum over all the vehicles and obtain a global collision risk (**lefevre2014surveye**).

2.3 Automotive Collision risk metrics

A risk metric in general means a parameter that is used to quantify the risk. This value would then help to choose the necessary methods or follow necessary steps to reduce the risk of the environment. In case of automotive domain, various risk metrics such as Time to Collision (TTC) help calculate the current risk of the situation.

The idea of computing a time-to-collision (TTC) was first suggested by Hayward (Hayward, 1972). He defined it as "the time required for two vehicles to collide if they continue at their present speed and on the same path." Hydén, in (Hydén, 1987) suggested that lower TTC values correspond to higher conflict severities. Although this point has been argued in the safety assessment literature, it seems clear that lower TTC values correspond to a higher probability of collision (Kruysse, 1991), (Tiwari, Mohan, and Fazio, 1998).

Hence, TTC is generally perceived to be a primary and efficient measure in traffic safety assessment especially in assessing conflicts. TTC is one of the most common safety surrogate assessment measures employed in microscopic simulation (Hou, List, and Guo, 2014). The different automotive collision metrics are discussed in the sections 2.3.1, 2.3.2 and 2.3.3.

2.3.1 Time to X (TTX) collision metrics

Situation analysis is an important capability of advanced driver assistance systems (ADAS) and has three main objectives: the representation and interpretation of the current traffic situation, the prediction of the dynamic evolvement of the scene, and criticality assessment (Feth, Schneider, and Adler, 2017).

Very convenient criticality measures are the time metrics, also known as Time-To-X (TTX). Time metrics characterize the criticality of the situation by time intervals until a predicted critical event occurs. Time metrics characterize the criticality of the situation from the point of view of the ego vehicle which is the vehicle with respect to which the risk metrics are calculated. Therefore they quantify the time intervals until a relevant critical event occurs (Berthelot et al., 2012).

Though widely used criticality measure is TTC, many other methods for threat assessment have been described in the literature (Hillenbrand, Spieker, and Kroschel, 2006), (Horst, 1991) which propose Time to React (TTR) as activation criterion for ADAS. Literatures like (Kaempchen, Schiele, and Dietmayer, 2009), (Ferguson et al., 2008) and many others propose TTR as an activation criteria for ADAS. TTR is the remaining time until the very last possible driving maneuver by which a driver can avoid an imminent collision. It can be approximated as the maximum of the time-to-brake (TTB), time-to-steer (TTS), and time-to-kickdown (TTK) in practice. And also, TTR reflects the ability of a vigilant driver to prevent an accident and can be interpreted easily in ADAS (Tamke, Dang, and Breuel, 2011).

Metric called Time to Maneuver in (Schreier, 2016) indicates the last moment in time when a maneuver achieves collision avoidance.

2.3.2 Worst time to Collision (WTTC)

In autonomous driving scenarios, if the goal is to not neglect any critical situations, a worst case estimation of possible uncertainties is required. There always exists a major uncertainty about the position of dynamic elements over time for road vehicles. This uncertainty results from physical possibilities due to the degrees of freedom and from the behavior of the conductor controlling the dynamic element (Wachenfeld et al., 2016).

As described in (Wachenfeld et al., 2016), WTTC aims to get a worst case approximation of all situations, thereby neglecting as few critical situations as possible which means that even considering the worst case actions, all dynamic objects are out of scope.

The most simple and comprehensive model covering all possible movements of a vehicle is Kamm's circle. One of the most important assumption by (Wachenfeld et al., 2016) for WTTC is that the metric does not differentiate situations by their potential severity of accident but rather identifies every potential accident to be avoided.

Kamm's circle is commonly known as circle of forces. When a car is driving on a straight line or during cornering with a constant velocity, only pure longitudinal or lateral forces will be transmitted. In general the longitudinal and lateral slip are overlapped (Schramm, Hiller, and Bardini, 2014). During a driving situation, the car is considered as a mass and an imaginary circle with a required radius is assumed to be present around the car. This concept is extended for collision prediction in (Wachenfeld et al., 2016) where, the kamm's circle is further modified to cater the needs required to solve a given problem. By introduction of yaw angle and parameterizing the maximum acceleration, more realistic driving behavior is reached. Whenever a car enters this imaginary circle, collision risk metric is calculated and also information could be given to collision avoidance systems to give a warning to the driver about the other vehicle.

However, the WTTC correlates with the severity as, higher relative velocities lead to smaller WTTCs and higher severities of accidents. Also, a small WTTC is not necessarily connected with a severe accident. This makes clear that the WTTC is not suitable to assess the criticality of a situation on its own but, requires subsequent metrics for the criticality assessment (Wachenfeld et al., 2016).

2.3.3 Other safety metrics

According to (Schreier, 2014), two major challenges exist in the design of long-term trajectory prediction and criticality assessment algorithms for active collision avoidance and warning systems. First, it is neither optimal to determine just a single future trajectory for each vehicle nor is it reasonable to predict every physically possible trajectory. In the first case, the one and only future hypothesis will most certainly not occur whereas human drivers take different scene evolutions into account.

In the second case, false warnings will be generated. The second less considered challenge is that the further one tries to predict into the future, the more assumptions have to be made, which tempts to model the average, sensible driver in a given traffic situation (Schreier, 2014).

The approach in (Schreier, 2014) connects the qualitative maneuver detection with the quantitative trajectory prediction domain with the ultimate goal to calculate a criticality measure suitable for arbitrary, uncertain driving environments for longer prediction time spans – the so-called Time-To-Critical-Collision-Probability (TTCCP).

Other metrics or safety indicators for example are:

- Deceleration rate,
- Safe stopping distance,
- Time required to conflict zone (Dijkstra and Drolenga, 2008),
- Potential Collision Energy (PCE) which is built up from the weights and speeds of the vehicles involved and the way in which they collide, indicates how much energy is released in the event of a collision between the vehicles that are in conflict with each other (Dijkstra and Drolenga, 2008).
- Time Exposed Time to Collision (TET) indicates the length of the time that a vehicle's TTC is below a critical value during a certain period. TET therefore is the sum of the moments that a vehicle has a TTC that is below that critical value. A disadvantage of TET indicator is that any TTC value that is lower than the critical value is not included in the calculation hence, TIT was developed (Dijkstra and Drolenga, 2008). It is more used for retrospective analysis.
- Time Integrated Time to Collision (TIT) calculates the surface area between the critical value and the TTC that occurs (Dijkstra and Drolenga, 2008).
- One limitation of TTC is that it assumes constant vehicle velocity. Enhanced Time to Collision (ETTC) is potentially a more accurate metric of perceived collision risk due to its consideration of vehicle acceleration (Chen, Sherony, and Gabler, 2016).
- Distance headway (DH) is the distance between a vehicle and the vehicle in front of it. Further indicators based on DH are Time Exposed Distance Headway (TEDH) and Time Integrated Distance Headway (TIDH) (Dijkstra and Drolenga, 2008).
- Time Headway introduced by Vogel (Vogel, 2003) is the time between a vehicle and the vehicle in the front.
- Criticality index divides speed by TTC.
- Time to Avoidance (TTA) (Jiménez, Naranjo, and García, 2013)

- Time Exposed Speed (TES) indicates the length of time that a vehicle's speed is above the speed limit for a road section during a certain time period (Dijkstra and Drolenga, 2008).

Few popular indicators of the criticality of a potential collision also include the velocity of the vehicles, the amount of overlap between the shapes representing the vehicles, the probability of simultaneous occupancy of the conflict area by both vehicles, and the configuration of the collision.

2.4 Safety Certificates for Open Adaptive Systems

(Schneider and Trapp, 2013b) points out that the systems used for autonomous tasks are safety critical, openness and adaptivity must not impede safety assurance. These systems are expected to be adaptive in a way that enables them to react appropriately to dynamic changes in device/service availability, resource availability, system environment, or user requirements. Current safety assurance approaches, however, presume that the system is completely specified and configured at design time. This is difficult to attain for adaptive systems and impossible for open systems. Since traditional approaches are not expected to scale to adequately address open adaptive systems, innovative safety assurance concepts are required as a key enabler for unfolding the promising economic potential of open adaptive systems.

Shifting some parts of the safety assurance activities into runtime when all open variables can be resolved and the safety-related state space is correspondingly much smaller than at development time could be a general solution approach. However, the aim must be to minimize the amount of safety assurance responsibility that is given to the system. As a possible solution, the concept of conditional safety certificates (ConSerts) was introduced in (Schneider and Trapp, 2013b).

According to (Schneider and Trapp, 2013a), ConSerts are predefined modular safety certificates of the single systems (i.e, devices) that are to be integrated. Modular certificates are based on the idea of contracts. Safety guarantees are defined that are fulfilled by a component under certain preconditions. First, the components define a set of safety demands, that is, safety requirements that must be fulfilled by the integration context. Second, they define invariants expressing assumptions made during the safety assurance of the component which must also be fulfilled by the integration context (Schneider and Trapp, 2013a).

ConSerts are thus not static but conditional. Conditions within a ConSert manifest themselves in the relations between potentially guaranteed safety requirements (Guarantees) and corresponding demanded safety requirements (demands) (Schneider and Trapp, 2013a), more about which is discussed in 3.

Chapter 3

Concept of dynamic risk assessment at the intersection

This chapter throws a light on the various concepts that are necessary to carry out the goal of performing a Dynamic Risk Assessment (DRA) at the intersection. Section 3.1 describes the steps to be followed to build an intersection using SUMO simulator. Section 3.2 tells about the prediction models that will be used for the current work. Section 3.3 introduces a safety world model and its relation with DRA. Sections 3.4 and 3.5 explain how the risk metrics are calculated and introduce a new risk metric called MTTC. Sections 3.6, 3.7 and 3.8 give information about ConSerts such as the demands and guarantees that are to be considered and how they are going to be used in the current work. In section 3.9 the different vehicle types which are going to be used are defined. The last section 3.10 gives an introduction to Risk Metric Calculator (RMC) which is used for calculation of the risk metrics and to perform DRA.

3.1 Intersection scenario considered

The intersection that is considered for the current thesis work is a four way signalised intersection each way having two lanes, one for the incoming vehicles i.e, vehicles moving away from the intersection and one for the outgoing vehicles i.e, the vehicles moving closer towards the intersection. The movement of the vehicles, i.e, the route which the vehicles take are defined in the route.xml file. There are a total of eight detectors, four detectors each placed on the incoming lanes and other four detectors each placed on the outgoing lanes. The incoming detectors detect the vehicle that entered the loop and the outgoing detectors remove the vehicles from the loop once they have passed over them as in the figure 3.1.

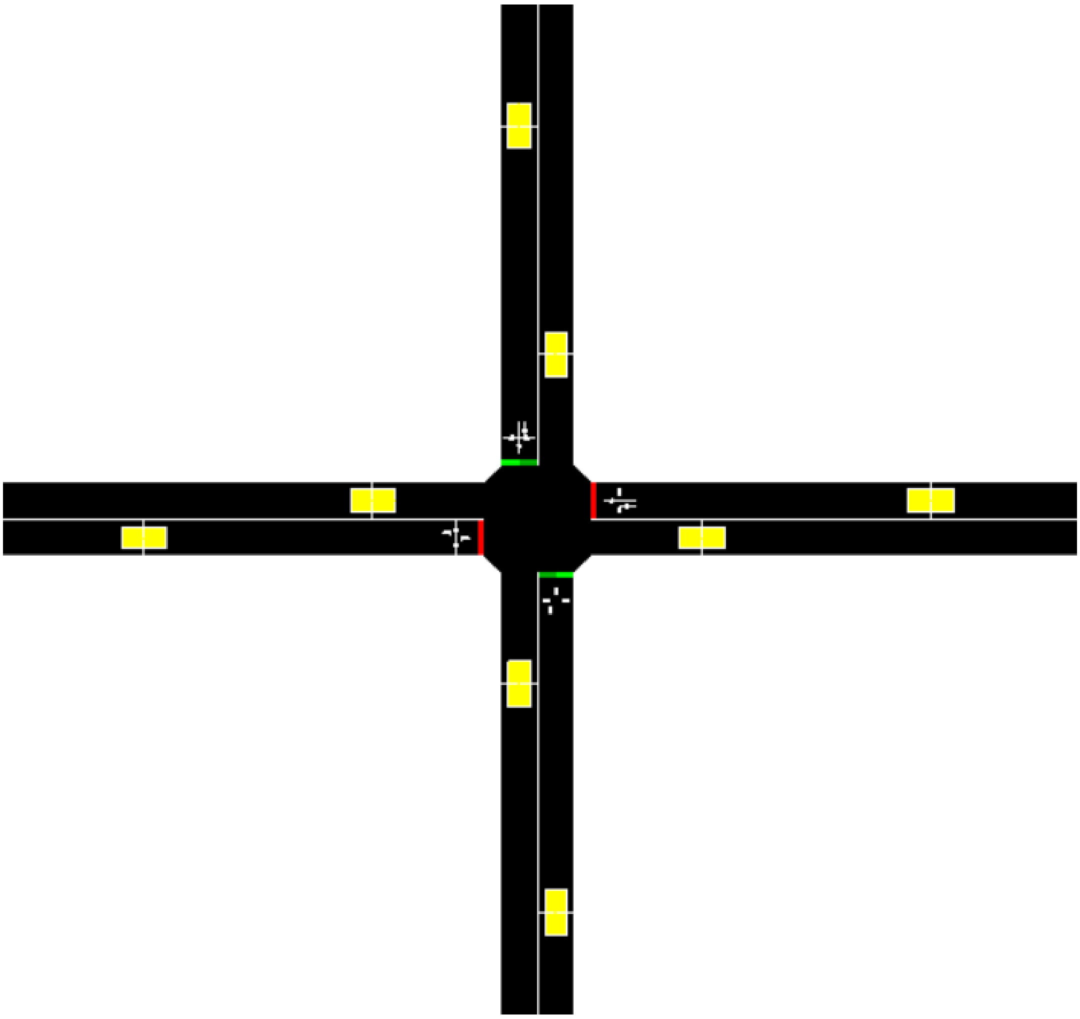

FIGURE 3.1: Intersection in SUMO with detectors as yellow rectangles

In SUMO, a street network consists of nodes (junctions) and edges (streets connecting the junctions). Thus, if it is necessary to create a network with two streets subsequent to each other, three nodes and two edges are needed. By following the steps in the process chart as in the figure 3.2, the intersection scenario is built for simulation in SUMO simulator.

FIGURE 3.2: Process chart showing steps involved in building a scenario using XML for simulation using SUMO simulator

3.2 Prediction models for the intersection

The prediction models that are being used for the current scenario i.e,. the crossing of an intersection are trajectory prediction model and reachable area prediction model. The reachable area uses Kamm's circle as discussed in the section 2.3.2, with different confidence levels that are defined in the section 3.8.

Once the scenario is fixed i.e. the number of vehicles with vehicle types defined, the probability distribution of the the vehicles, the routes that the vehicles are intended to take, etc,. the whole simulation is run using SUMO simulator. The trajectory information for all the vehicles is stored in a separate file. How the information about trajectory is extracted and stored for each vehicle is discussed in detail in chapter 4.

3.3 Dynamic Risk Assessment and Safety World Model

In the automotive domain, commands such as steering, accereleration and deceleration are produced with respect to systems of higher automation level. For the current driving situation, these commands need to be sufficiently adequate. One of the core aspects of today's development towards higher automation levels is the needed situation awareness. In order to achieve a safe behavior, the aspect of the current risk of situation is an important aspect of the overall situation awareness. Hence approaches to create situation awareness towards risk are called Dynamic Risk Assessment (DRA).

Also, DRA is performed by the calculation of risk metrics. A huge set of automotive collision risk metrics are available with different limitations and assumptions. An active safety system for an autonomous vehicle needs to assess the risk of all possible accidents and not just a small subset of it. Hence, depending on the current driving situation, evaluation of different risk metrics has to be done.

A Safety World Model (SWM) is the collection of models and assumptions that enable to perform a Dynamic risk assessment (Feth, Schneider, and Adler, 2017). Figure 3.3 represents the Safety World Model. This model contains an internal representation of the current environment as:

- Situation Description which is an understanding of how the current situation may evolve in the future
- situation Prediction
- Situation Risk which is an assessment of the risk of the current situation.

These three elements are dependent on each other. The output is a decision about whether the normal control algorithm is allowed to control the system further or whether some countermeasures are needed to steer the system into a safer state (Feth, Schneider, and Adler, 2017).

FIGURE 3.3: Safety World Model (Feth, Schneider, and Adler, 2017)

According to (Feth, Schneider, and Adler, 2017), a genuine Dynamic Risk Assessment (DRA) thus builds a Situation Description containing all relevant elements of the current operational situation, a Situation Prediction that guaranteed to not miss any potential unwanted events in the future and thus rather overestimated the possible future situation space and a Situation Risk Assessment that explicitly considers the probabilities and severities of unwanted events in the predicted situations.

3.4 Risk metric for dynamic risk assessment

Risk metrics are domain specific. It is a challenge to choose the right metrics for dynamic risk assessment for a domain. The risk of a situation can be assessed qualitatively as in (Mekki-Mokhtar et al., 2012), or quantitatively, as in (Nunen et al., 2016). Special safety metrics are needed if the risk of a situation is assessed quantitatively. Such special safety metrics rate the criticality of a situation, which is the risk that it may result in a harmful situation. (Feth, Schneider, and Adler, 2017) mentions that a risk assessment is needed to separate the situation space that the system might encounter in a safe space, where the complex control algorithm is allowed to operate, and a potentially unsafe space, where actions by a Safety Supervisor are needed to keep accidents from happening. The space of possible metrics is limited by the attributes considered in the Situation Description and the prediction models in the Situation Prediction.

As discussed in the previous chapters, collision avoidance is the main and the most important criteria for autonomous vehicles with respect to safety. Among the various risk metrics that are being used for risk assessements, Time to Collision (TTC) would be the risk metric with respect to each vehicle that would be considered as a base to calculate the metrics such as General Time to Collision (GTTC), Intersection Time to Collision (ITTC) in order to arrive at a global level risk metric for the intersection for dynamic risk assessment for the current scenario. The metrics are calculated with the Risk Metric Calculator (RMC) by using a grid map which is explained in detailed in section 3.10. Time to collision has to be calculated for those vehicles which have futuristic cell prediction because in case of vehicles that aren't moving, calculating the time to collision would not be appropriate as there is a chance that it always shows as the most critical scenario.

Each vehicle has different time to collision values with respect to every other vehicle that are inside the current loop as detected by the incoming detectors in SUMO simulator. General Time to Collision metric consists of these values of the TTC metric for each vehicle considering multiple target vehicles which is local to the vehicle for a particular time step.

For example, if there are 3 vehicles veh1, veh2, veh3 at the intersection for a particular time step, each vehicle will have two GTTC values. Mathematically, the number of GTTC values say N for each vehicle will be:

$$N = Number\ of\ vehicles\ at\ the\ intersection - 1$$

Intersection Time to Collision metric coniders multiple vehicles. But it still calculated for every vehicle which is the minimum of all the GTTC values for a vehicle for a single time step. Each vehicle has one value for ITTC. But this is not sufficient to estimate or perform a dynamic risk assessment for the overall system. Hence a new metric Mission Level Time to Collision (MTTC) is defined for mission level dynamic risk assessment which is further explained in section 3.5.

Considering the same example as above for three vehicles at the intersection, the ITTC value for veh1 would be minimum of (GTTC_1,GTTC_2). This can mathematically be represented as:

$$ITTC_{vehicle} = \min GTTC_{vehicle}$$

3.5 Mission level time to collision - MTTC

Each moving vehicle has multiple GTTC values with respect to other moving vehicles near the intersection depending on the prediction model related to the vehicle type. To arrive at ITTC for each vehicle, the minimum value of all the GTTC's is considered for each vehicle. The minimum of all the ITTC's is considered as the MTTC for the intersection. But there can arise situations where the MTTC value can be same for two different scenarios or time steps.

For example, out of three vehicles, if veh1 has GTTC values 2 secs and 3 secs, veh2 has GTTC values 2secs and 4 secs, and veh3 has GTTC values 3 secs and 4 secs, the ITTC value for veh1 would be 2 secs, for veh2 would be 2 secs and for veh3 it would be 3 secs at the intersection. veh1 and veh2 have a common value of ITTC but, the criticality is not the same. veh1 GTTC values are more critical compared to veh2. In such a case, though we are considering a worst case minimum value as ITTC, to distinguish the criticality of the scenarios, an additional parameter is used. Mathematically:

$$MTTC_{intersection} = \min ITTC_{vehicle}$$

$$AVG = \frac{GTTC_{of\ all\ vehicles}}{Total\ number\ of\ vehicles\ at\ the\ intersection}$$

The additional parameter AVG calculates the average of all the GTTC values of all vehicles at the intersection for that particular time step. Both ITTC and AVG values can then be used for differentiating critical scenarios. Hence, MTTC remaining same, lower the AVG value, higher is the criticality.

3.6 ConSerts

The definition of ConSerts relies on the existence of a domain-wide standardization of service types and safety property types (Schneider and Trapp, 2013b). According to (Schneider and Trapp, 2013b) ConSerts define a safety contract that can be evaluated at runtime. In order to use these ConSerts at runtime, a certain level of formality is required. To formalize its guarantees and demands and to specify them in an unambiguous way for a given domain is the first step. To this end, safety guarantees are defined, which are safety requirements that can be fulfilled by one system under the assumption that its safety demands are fulfilled by the integration context.

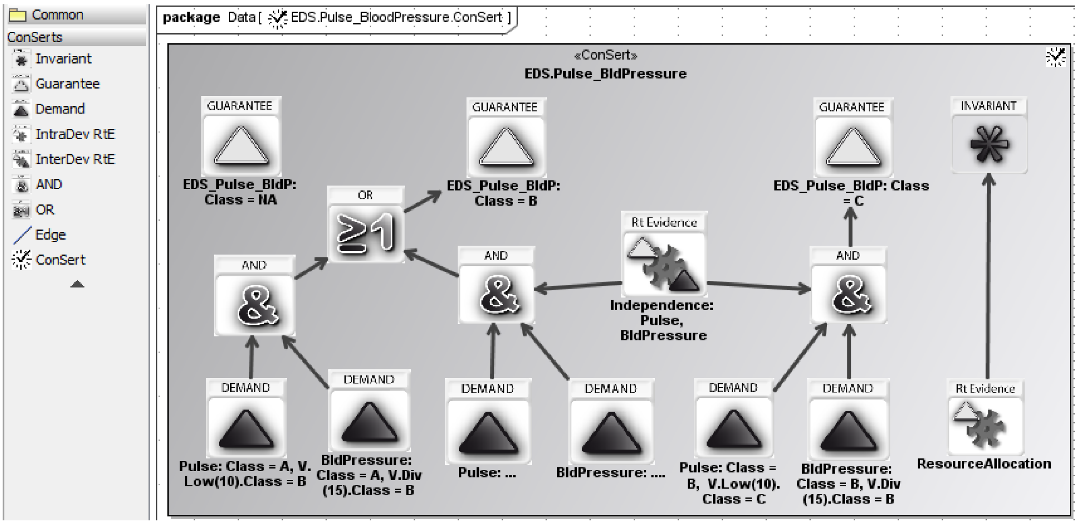

FIGURE 3.4: Model example for ConSerts (Schneider and Trapp, 2013a)

It is generally necessary to be able to certify that a system guarantees certain safety properties under certain conditions in order to apply ConSerts for self-adaptive systems (Schneider and Trapp, 2013a). Figure 3.4 represents a model describing a ConSert for EDS configuration EDS.Pulse BldPressure. It shows the various elements such as gates, demands, guarantees, run time evidences that are used for defining ConSerts. A detailed explanation can be seen in (Schneider and Trapp, 2013a).

3.7 Architecture ConSert tree

As for interoperability, open systems generally require some formal concept to abstract from concrete component interfaces. In order to dynamically negotiate compositions, it is essential to formally describe which functional and non functional property components provide and what is required by the component for their realization (Schneider and Trapp, 2013a). Architecture models, help define how the components are connected. Various existing architectures such as Service oriented architecture (SOA), integrated architecture, etc,. can be used according to the need for the system. In a simpler way, the architecture being used currently for the thesis, can be represented in the form of a ConSert tree as in the figure 3.5.

Dynamic risk assessment is performed for three confident levels - high, mid and low. Respective services have to be fulfilled in order to perform the risk assessment. The risk of the intersection scenario can be based on two different prediction models – reachable area and intended trajectory. The intended trajectory risk assessment would require the demands for intended trajectory and Vehicle Position to be guaranteed from the vehicle side. The reachable area risk calculation would need Vehicle Position, Vehicle Speed and yaw angle demands to be guaranteed with the same confidence level. The infrastructure inputs are used in case of a low confidence model when no other data from the vehicle is available, for example the vehicle type Legacy defined in the section 3.9 and also in case of mid confidence level. Thus, depending on the scenario and driving situation, the risk assessment can be carried out.

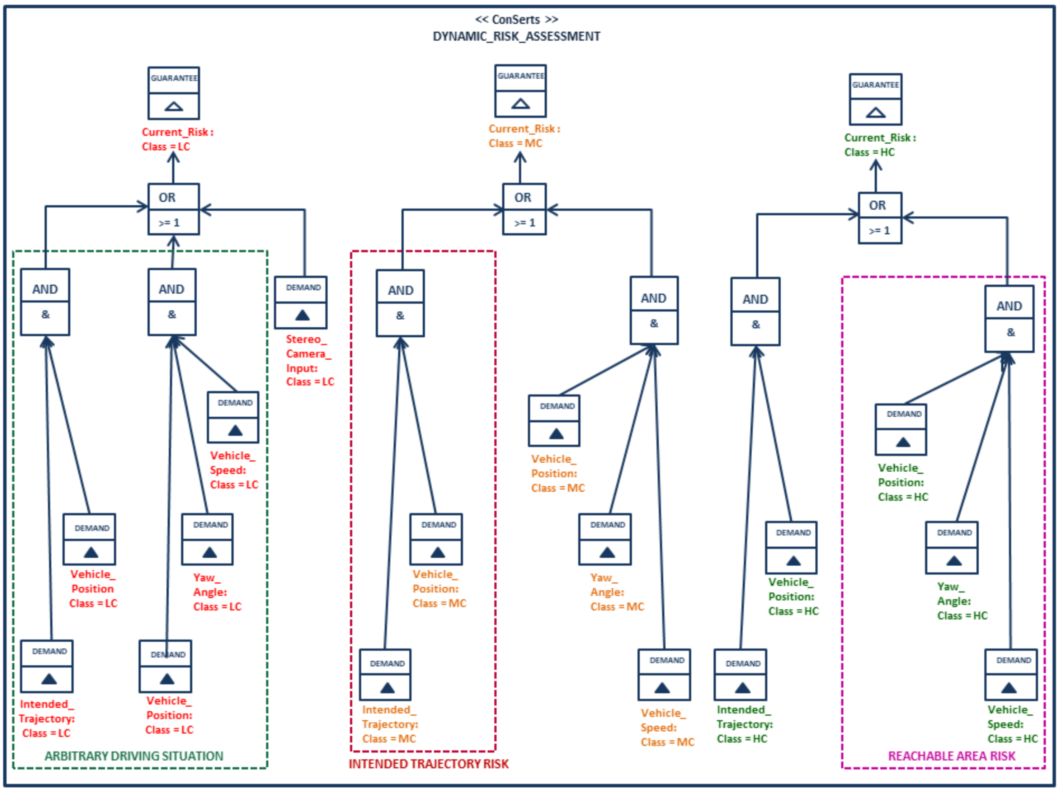

FIGURE 3.5: ConSert tree for architecture considered for ConSerts evaluation, visualization and dynamic risk assessment

3.8 Demands and guarantees for ConSerts

Demands always represent the safety requirements relating to the environment of a component, which consequently cannot be verified at design time already (usually addressed with worst case assumptions) which in other words represent requirements with respect to the safety properties of an external service. A ConSert therefore certifies that the guarantees will hold with acceptable probability under the precondition that the specified safety demands are fulfilled by the environment (Schneider and Trapp, 2013a). Safety properties can also be used to specify the guarantee or demand that is assured with a specific level of confidence that certain safety requirements will be satisfied (Schneider and Trapp, 2013b).

The services that will be used to define the demands and guarantees for the current thesis work are:

- Dynamic risk assessment
- Vehicle position

- Vehicle trajectory
- Vehicle yaw
- Vehicle speed
- Infrastructure (stereo camera for example)

  Also, the three different confidence levels used are:

- High confidence (HC)
- Mid confidence (MC)
- Low confidence (LC)

  All these services associated with a confidence level are used to define different vehicle types which are explained in section 3.9 and also to come up with the architecture as in section 3.7. Infrastructure does not have any confidence level associated with it.

Neutral (i.e., neither fulfilled nor not fulfilled) ConSerts where the demands are guaranteed from the vehicles perspective with respect to each confidence level can be seen in the figure 3.6.

3.9 Vehicle types based on the services considered for ConSerts

The vehicle types are defined based on the capabilities of the vehicles. Those capabilities translate to guarantees. The vehicles defined for the scenario being considered for the thesis are of four types:

- "All_HC": – stands for All High Confidence. It is of the type where the demands, vehicle position (VP), vehicle yaw (VY), vehicle speed (VS), vehicle trajectory (VT) are met with high confidence. It makes it possible to calculate intended trajectory risk and also reachable area risk. The neutral guarantees for the vehicle type ALL_HC are seen in the figure 3.7.

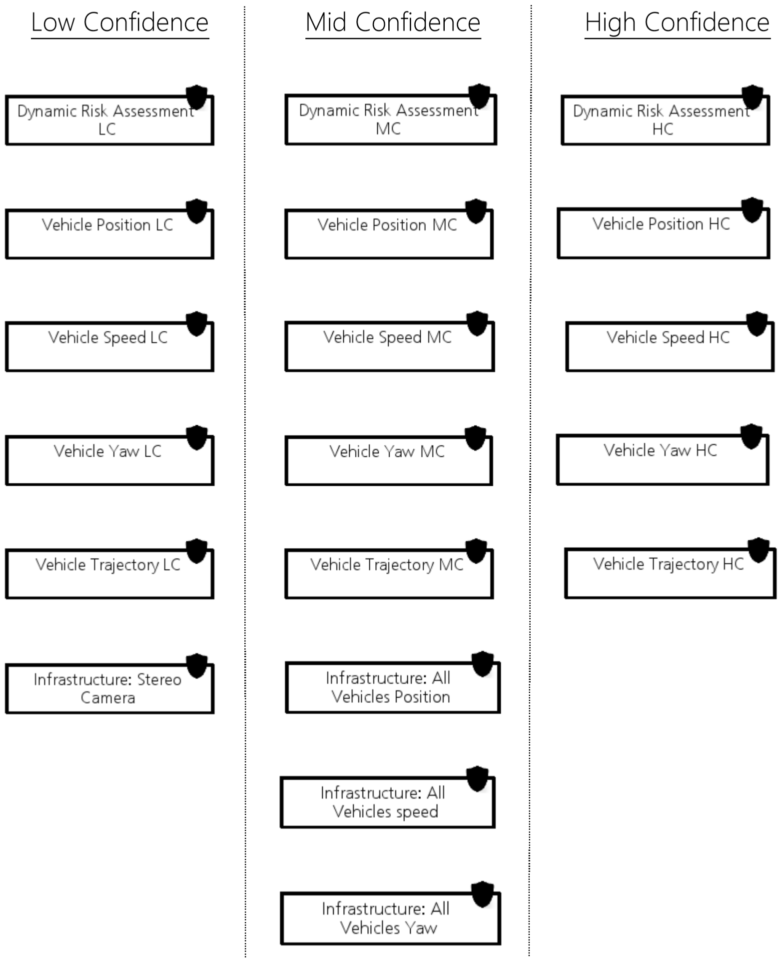

FIGURE 3.6: All possible guarantees, the vehicle types have a subset of it

- "RA_HC": – stands for Reachable Area High Confidence. It is of the type whose demands vehicle position (VP), vehicle yaw (VY), and vehicle speed (VS) are met with high confidence, making it possible to calculate reachable area risk. The neutral guarantees for the vehicle type RA_HC are seen in the figure 3.8.

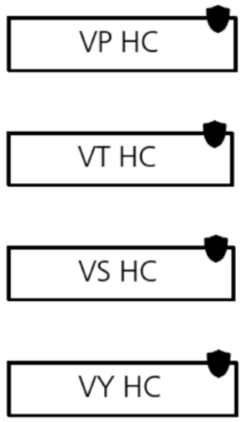

FIGURE 3.7: Neutral guarantees for the vehicle type All_HC

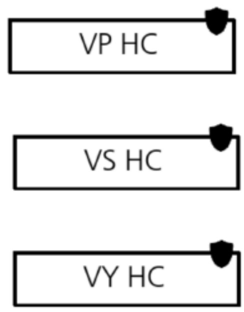

FIGURE 3.8: Neutral guarantees for the vehicle type RA_HC

- "IT_MC": – stands for Intended Trajectory Mid Confidence. It gives guarantees from the vehicle's perspective with the demands vehicle position (VP), vehicle yaw(VY), vehicle speed (VS) and vehicle trajectory (VT) met with mid confidence. The neutral guarantees for the vehicle type IT_MC are seen in the figure 3.9.

- "Legacy": – represents vehicles which would not support any V2V or V2I communication. has no demands met with any confidence i.e, doesn't give any guarantees. It only contains the information from the infrastructure (i.e., stereo camera (SC) for example) which will be the input with respect to the vehicle. High confidence and mid confidence risk assessment is not possible in this case. Only low confidence risk assessment can be performed. The neutral guarantees for the vehicle type Legacy are seen in the figure 3.10.

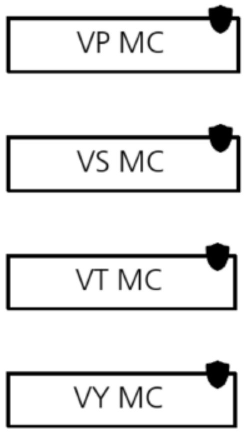

FIGURE 3.9: Neutral guarantees for the vehicle type IT_MC

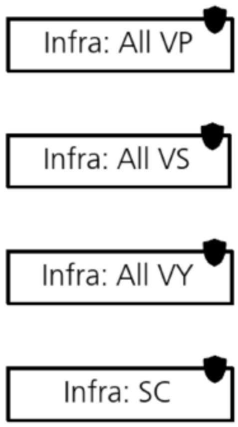

FIGURE 3.10: Neutral guarantees for the vehicle type Legacy

3.10 Risk Metric Calculator (RMC)

Risk Metric Calculator (RMC) is the tool used for the development and comparison of approaches for Dynamic Risk Assessment of active safety systems. The RMC is written is python and has a minimalistic GUI which can be used for evaluating some pre configured DRA approaches. The tool is mainly used by extending the python scripts directly.

For creating the scenarios in which Dynamic Risk Assessment is evaluated, one method is to directly define the scenario in a python script specifying necessary information such as initial position, acceleration values, etc, once for each vehicle. The other method is to use simulators for specification of scenarios with the possibility to use the visualization of simulator for later analysis.

Once the scenario has been imported into the Risk Metric Calculator, the information is translated to a Grid Map, in which the world is discretized in fixed cells of 10 * 10cm. For each of those cells and for each step in the scenario, the RMC calculates the occupancy information. This Grid Map is then used for calculation of risk metrics. RMC contains different metrics such as time headway, general time to collision, etc,. resulting in different approaches for Dynamic Risk Assessment. To extract necessary information, work has to be done on the information related to the grids that are active. Each vehicle is represented as a set of points occupying a particular cell/grid. It is the vehicle occupancy that determines which cells/grids are active. RMC contains some visualization features because of which it is possible to visualize the imported scenarios and calculated Grid Map in a simplistic way. In the current work, a connection is established between SUMO simulator and RMC inorder to perform a DRA. The various functionalities used for this purpose are explained in detail in chapter 4.

Chapter 4

Implementation of Mission level Dynamic Risk Assessment and ConSerts at the intersection

For the implementation of the concepts explained in chapter 3 , Risk Metric Calculator (RMC) is used with its connection to the SUMO simulator and the ConSerts runtime environment. The intersection scenario considered in chapter 3 is simulated and in parallel, the mission level risk metric is calculated in the Risk Metric Calculator.

4.1 Implementation in SUMO simulator

The algorithm is written in the Risk Metric Calculator. As a first step, the vehicle types that have been discussed in 3.9 are to be defined in the route file of SUMO simulator using Python scripting language.The routes that the vehicles take have been decided to be East-to-West (EW), West-to-East (WE), North-to-South (NS), South-to-North (SN) and East-West-Turn-Right (EWTR) meaning EW vehicles turn right depending on the vehicle distribution for routes.

Also, each vehicle type is given a distinct color so that it would be easy to distinguish when the simulation is run. The vehicles of type IT_MC are given light blue color, type RA_HC are given pink, All_HC are given dark blue and red color is given to the vehicle type Legacy. Different vehicle distributions i.e, demand per second from different directions are then used to define and change the traffic density.

All the code in the figure 4.1 is written in the function generate_routefile().

```
print("""<routes>
 <vType id="RA_HC" accel="0.8" decel="6.5" sigma="0.5" length="5" minGap="2.5" maxSpeed="13" guiShape="passenger"/>
 <vType id="IT_MC" accel="0.8" decel="6.5" sigma="0.5" length="7" minGap="3" maxSpeed="13" guiShape="bus"/>
 <vType id="All_HC" accel="0.8" decel="6.5" sigma="0.5" length="7" minGap="3" maxSpeed="13" guiShape="bus"/>
 <vType id="Legacy" accel="0.8" decel="6.5" sigma="0.5" length="7" minGap="3" maxSpeed="13" guiShape="bus"/>


 <route id="WE" edges="51o 1i 2o 52i" />
 <route id="EW" edges="52o 2i 1o 51i" />
 <route id="SN" edges="53o 3i 4o 54i" />
 <route id="NS" edges="54o 4i 3o 53i" />
 <route id="EWTR" edges="52o 2i 4o 54i"/>
```

FIGURE 4.1: Vehicle types and routes defined in generate_routefile()

Now to establish the connection between SUMO simulator and the Risk Metric Calculator in order to build the occupancy map from the SUMO simulation results, three data structures are passed to the RMC as parameter, which are as follows:

- managed_vehicles [ ] holds all vehicles for which, the dynamic_vehicle_ information and static_vehicle_information and is currently maintained by vehicle IDs.

- static_vehicle_information [ ] contains static information about a managed vehicle. Data structure is a dictionary with the vehicle id as the key to an entry. Entries are structured as follows:

  [length, width, low confidence trajectory, mid confidence trajectory, high confidence trajectory]

- dynamic_vehicle_information [ ] contains dynamic information about a managed vehicle. Data structure here is a dictionary with the vehicle id as the key to an entry. Entries are structured as follows:

  [pos_x, pos_y, speed, -orientation, prediction model low confidence, prediction model mid confidence, prediction model high confidence]

Whenever the vehicles are detected by the incoming detectors, the information is added to the required data structures and when the vehicles pass through the outgoing detectors, the information about the vehicles is removed from the necessary data structures for every time step. Static information i.e, information which does not change with time is retained, for example the length of the vehicle for a particular vehicle type.

```xml
<data timestep="1.00">
    <vehicles>
        <vehicle id="All_HC_1" eclass="HBEFA3/PC_G_EU4" CO2="2687.41" CO="152.50" HC="0.76" NOx="1.21"
         PMx="0.06" fuel="1.16" electricity="0.00" noise="59.68" route="NS" type="All_HC"
         waiting="0.00" lane="54o_0" pos="7.88" speed="0.78" angle="180.00" x="508.35" y="1012.12"/>
        <vehicle id="All_HC_2" eclass="HBEFA3/PC_G_EU4" CO2="2620.00" CO="156.92" HC="0.78" NOx="1.19"
         PMx="0.06" fuel="1.13" electricity="0.00" noise="58.06" route="WE" type="All_HC"
         waiting="0.00" lane="51o_0" pos="7.54" speed="0.44" angle="90.00" x="7.54" y="508.35"/>
        <vehicle id="RA_HC_0" eclass="HBEFA3/PC_G_EU4" CO2="2629.12" CO="155.88" HC="0.77" NOx="1.19"
         PMx="0.06" fuel="1.13" electricity="0.00" noise="58.40" route="EWTR" type="RA_HC"
         waiting="0.00" lane="52o_0" pos="5.62" speed="0.52" angle="270.00" x="1014.38" y="511.65"/>
    </vehicles>
    <edges>
        <edge id=":0_0" traveltime="0.33">
            <lane id=":0_0_0" CO="0.00" CO2="0.00" NOx="0.00" PMx="0.00" HC="0.00" noise="0.00"
             fuel="0.00" electricity="0.00" maxspeed="15.28" meanspeed="55.01" occupancy="0.00" vehicle_count="0"/>
        </edge>
        <edge id=":0_1" traveltime="0.62">
            <lane id=":0_1_0" CO="0.00" CO2="0.00" NOx="0.00" PMx="0.00" HC="0.00" noise="0.00"
             fuel="0.00" electricity="0.00" maxspeed="15.28" meanspeed="55.01" occupancy="0.00" vehicle_count="0"/>
        </edge>
```


FIGURE 4.2: Information available in the generated dump file

As discussed in chapter 3, having deterministic scenarios is a prerequisite. As a first step, the needed trajectory information is extracted by fixing the amount of time the simulation could be run for the considered intersection scenario. SUMO has an option to create a net - dump file for the simulation that runs. The produced XML Structure contains information about the edges, lanes, vehicles and traffic lights. The intention for this option was to check the simulation results without having to record all the simulation commands e.g. traci. This file is used to extract the trajectory information.

To force SUMO to build a file that contains the full dump, the command line is extended by the parameter –full-output <FILE>. <FILE> is the name of the file to which the output will be written to. Any other file with this name will be overwritten, the destination folder must exist. output.xml is used as the FILE name and "–full-output", "output.xml" is added to the command line to generate the dump file. Every time the simulation is run, the output.xml file is rewritten, which makes it impossible to extract future information for vehicles. Hence,once the dump file is generated, it is copied and stored under a diffrent file name as "output_saved.xml" which looks like 4.2. This file is now used for extracting the future trajectories for the vehicles.

The dump file output_saved.xml has lot of information that might not be relevant. Hence, as it is an xml file, it can be parsed to only read the necessary information. xml.etree.cElementTree is used to parse the file. To this end, a function definition called input_for_trajectory (vehicle_ID, time_step, length, file) where vehicle ID, corresponding time step , length of the vehicle and file name that are taken

```python
def add_vehicles_to_managed_vehicles(vehicles):
    global static_vehicle_information
    global managed_vehicles
    for vehicle in vehicles:
        if not vehicle in managed_vehicles:
            managed_vehicles.append(vehicle)
            # get static information for that vehicle
            type_id = traci.vehicle.getTypeID(vehicle)
            # size in cells (1 dcm step)
            length = math.ceil(traci.vehicletype.getLength(type_id) * 10)
            width = math.ceil(traci.vehicletype.getWidth(type_id) * 10)
            static_vehicle_information[vehicle] = [length, width]
            # retreive and store the trajectory as part of the static_vehicle information
            current_time = traci.simulation.getCurrentTime() / 1000
            vehicle_saved_points = input_for_traj(vehicle, current_time, length, "output_saved.xml")
            #print("Saved points for vehicle: "+str(vehicle) + " :" +str(vehicle_saved_points))
            vehicle_trajectory_low_conf = construct_trajectory_from_saved_points(vehicle_saved_points, width, 0)
            #print("Constructed low conf trajectory for vehicle: "+str(vehicle) + " : "+str(vehicle_trajectory_low_conf))
            static_vehicle_information[vehicle].append(vehicle_trajectory_low_conf)
            vehicle_trajectory_mid_conf = construct_trajectory_from_saved_points(vehicle_saved_points, width, 1)
            #print("Constructed mid conf trajectory for vehicle: "+str(vehicle) + " : "+str(vehicle_trajectory_mid_conf))
            static_vehicle_information[vehicle].append(vehicle_trajectory_mid_conf)
            vehicle_trajectory_high_conf = construct_trajectory_from_saved_points(vehicle_saved_points, width, 2)
            #print("Constructed high conf trajectory for vehicle: "+str(vehicle) + " : "+str(vehicle_trajectory_high_conf))
            static_vehicle_information[vehicle].append(vehicle_trajectory_high_conf)
```

FIGURE 4.3: Code showing the information regarding the data structures managed_vehicles [ ] and static_vehicle_information [ ]

```python
def input_for_traj(vehicle_ID,time_step,file):
    center_x = 5000
    center_y = 5000
    offset_value_x = 400
    offset_value_y = 400
    origin_x = center_x - offset_value_x
    origin_y = center_y - offset_value_y
    tree = ET.parse(file)
    root = tree.getroot()
    Trajectory_info = []
    for timestep in root.findall("./data"):
        attrib_ts = timestep.attrib['timestep']
        ts = int(float(attrib_ts))
        for child in timestep:
            for vehicle in child:
                if vehicle.tag == 'vehicle':
                    if vehicle.attrib['id'] == vehicle_ID:
                        if ts >= int(float(time_step)):
                            Temp = []
                            Temp.append(ts)
                            Temp.append(vehicle.attrib['speed'])
                            Temp.append(vehicle.attrib['x'])
                            Temp.append(vehicle.attrib['y'])
                            Temp.append(vehicle.attrib['angle'])
                            Trajectory_info.append(Temp)

    return Trajectory_info
```

FIGURE 4.4: input_for_traj() function

as inputs is defined. The function returns trajectory_info.

The trajectory_info is then used to pass the information to static_vehicle_ information[ ] with different confident levels. Current time is calculated by using traci.simulation. getCurrentTime( ) and the file "output_saved.xml" is given as input. The code for managed_vehicles [ ] data structure and static_vehicle_information [ ] data structure is in the figure 4.3.

With possible values for confidence levels as 0 for low confidence, 1 for mid confidence and 2 for high confidence, trajectory is approximated by the borders of the trajectory format of trajectory. It has the list of entries with the structure:

[time, position x, position y, speed, orientation]

```python
def update_vehicles_dynamic_vehicle_information():
    list_of_vehicle_ids = []
    global prediction_model_low_conf
    global prediction_model_mid_conf
    global prediction_model_high_conf
    for vehicle in managed_vehicles:
        type_id = traci.vehicle.getTypeID(vehicle)
        pos_x, pos_y = traci.vehicle.getPosition(vehicle)
        length = static_vehicle_information[vehicle][0]
        # translation from vehicle position point from the front bumper to the center of the vehicle using x/y offset
        orientation = traci.vehicle.getAngle(vehicle)
        y_offset = length / 2 * math.cos(math.radians(360) - math.radians(orientation))
        x_offset = length / 2 * math.sin(math.radians(360) - math.radians(orientation))
        pos_x = math.ceil(10 * pos_x - origin_x + x_offset)
        pos_y = math.ceil(10 * pos_y - origin_y - y_offset)
        speed = traci.vehicle.getSpeed(vehicle)
        if type_id == "Legacy":
            prediction_model_low_conf = 3
            prediction_model_high_conf = 6
            prediction_model_mid_conf = prediction_model_low_conf
        elif type_id == "IT_MC":
             prediction_model_mid_conf = 5
             prediction_model_high_conf = 6
             prediction_model_low_conf = prediction_model_mid_conf
        elif type_id == "All_HC":
             prediction_model_high_conf = 5
             prediction_model_low_conf = prediction_model_high_conf
             prediction_model_mid_conf = prediction_model_high_conf
        elif type_id == "RA_HC":
            prediction_model_high_conf = 3
            prediction_model_low_conf = prediction_model_high_conf
            prediction_model_mid_conf = prediction_model_high_conf
        dynamic_vehicle_information[vehicle] = [pos_x, pos_y, speed, -orientation, prediction_model_low_conf,
                                                prediction_model_mid_conf, prediction_model_high_conf]
```

FIGURE 4.5: Code showing information about dynamic_vehicle_information [ ] data structure

The necessary algorithm is written in the construct_trajectory_from_saved_points (vehicle_saved_points, width, confidence_level) with vehicle_saved_points from the add_vehicles_to_managed_vehicles ( ) definition, width of the vehicle and confidence level (0, 1 or 2) as inputs.

The information regarding the position, yaw angle for every vehicle is stored in the form of list of lists, for each time step. For example, consider the total simulation time to be 50 seconds. Assume that at time 3 secs, there are two vehicles at the intersection. These two vehicles now contain information about position, speed etc, that they are going to take in the next second which is extracted from the dump file. This information helps to plot the predicted trajectories for the vehicle.

The code for dynamic_vehicle_information [ ] data structure is in the figure 4.5. A value each is given to the prediction_model_low_conf, prediction_model_mid_conf and prediction_model_high_conf, for each vehicle. The values 3, 5 and 6 are used in the current work. The value 3 is for Kamm's circle (accelerated, modified) model, 5 is for Trajectory-based prediction model and number 6 indicates no prediction.

Using all the above information, the current cells and the predicted cells which are the occupied cells, with and without confidence are calculated. The code as to how the calculations are done is in the Python file Occupancy_Map.py of the RMC which also uses different functions of Online_Helper_Functions.py, which is also in the RMC.

The occupancy map is used to calculate risk metrics. The map only contains information from one simulation step. Prediction horizon is an important parameter considered throughout. It gives the information about which cells the car is going to occupy at each time step. The value of prediction horizon is set to 1000 in the current work.

4.2 Mission level ConSert implementation

In order to work with ConSerts, services which include demands and guarantees have to be first documented. The guarantees and demands with respect to each vehicle type and confidence level are documented in the form of images in figs folder of Python_Visualization folder that is inside Evaluation folder. GIMP tool is used to create the figures with the existing templates. GIMP stands for GNU Image Manipulation Program. It is a cross-platform image editor available for GNU/Linux, OS X, Windows and more operating systems.

A dictionary is defined in image_dict.py in Python_Visualization folder that takes three attributes to look up for the images. The first attribute being the id (string), second attribute corresponds to type (demand (is given a value 0), guarantee (1), runtime evidence (2)) and the third attribute about the status (not fulfilled / open (0), fulfilled (1), neutral (2)). Special entries are created for the "and" and "or" element. The image_dictionary looks as the figure 4.6. Any new demands/ Guarantees/ Runtime Evidences need to be registered in the image_dict.py time to time.

```
image_dictionary = {
    "and" : "figs/And_Element.png",
    "or" : "figs/Or_Element.png",
    "(Vehicle_Position_HC, 0, 1)" : "figs/DRA_Vehicle_Position_HC_Demand_Fulfilled.png" ,
    "(Vehicle_Position_MC, 0, 1)" : "figs/DRA_Vehicle_Position_MC_Demand_Fulfilled.png" ,
    "(Vehicle_Position_LC, 0, 1)" : "figs/DRA_Vehicle_Position_LC_Demand_Fulfilled.png" ,
    "(DRA_HC, 1, 1)" : "figs/DRA_HC_Guarantee_Fulfilled.png" ,
    "(DRA_MC, 1, 1)" : "figs/DRA_MC_Guarantee_Fulfilled.png" ,
    "(DRA_LC, 1, 1)" : "figs/DRA_LC_Guarantee_Fulfilled.png" ,
```

FIGURE 4.6: Sample code showing image_dictionary

The Apache Thrift software framework, for scalable cross-language services development, combines a software stack with a code generation engine to build services that work efficiently and seamlessly between C++, Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, JavaScript, Node.js, Smalltalk, OCaml and Delphi and other languages. Thrift is used in the current work to establish communication between ConSerts_RTE_Evaluation_Service, ConSerts_RTE_Visualization_Service and RMC.

The ConSerts Evaluation Engine has to create a special list of elements that represent a tree. The top node needs to be the first element in that list. IDs are always two digits. The function Get_Evaluation_Result_For_Visualization (vehicleTypes) with vehicle types (RA_HC, All_HC,IT_MC and Legacy) as input returns the result which is to build the ConSert trees.

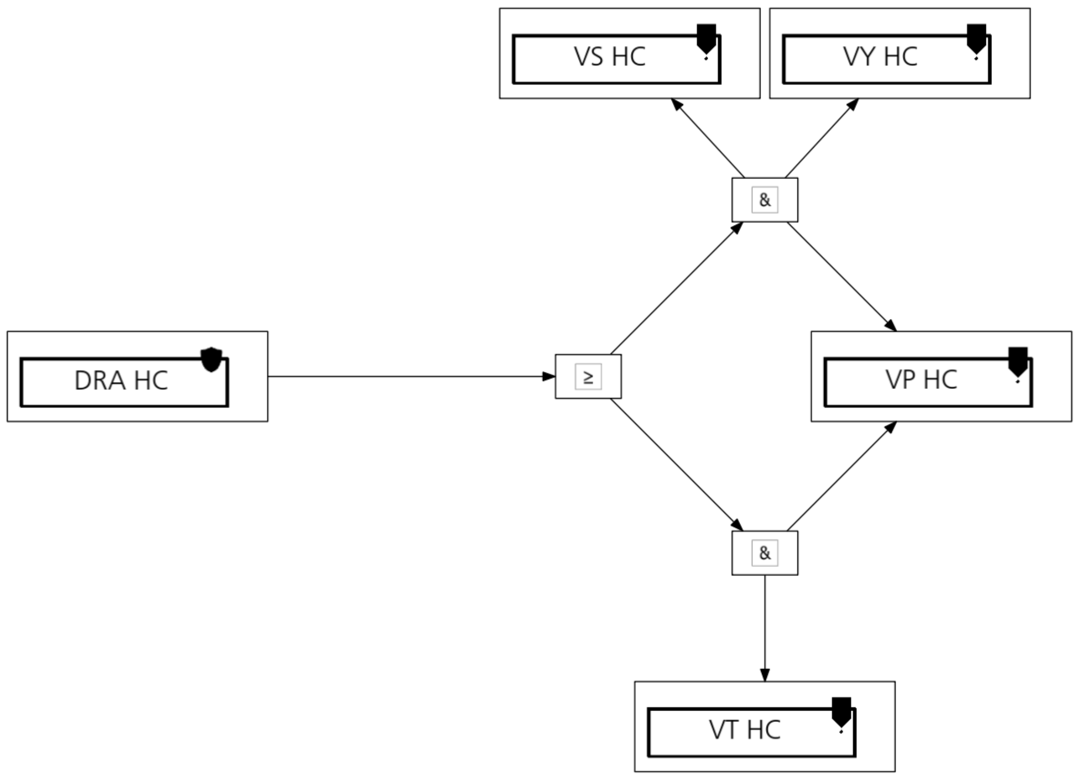

FIGURE 4.7: ConSert Tree for a high confidence Risk assessment with 1 vehicle

In the ConSerts_RTE_Visualization_Service.py, there is a function called Build _And_Render_Contract_Tree (tree_list) that takes a tree_list as an input. The tree list is created using the function Build_And_Render_DRA_Tree (conf_level, evaluation_result, vehicle_types) with the confidence level as the first input parameter, evaluation result being the second parameter and vehicle types being the third parameter.

In order to visualize the ConSert trees in a better way, Graphviz is used. Graphviz is an open source graph visualization software. Graph visualization is a way of representing structural information as diagrams of abstract graphs and networks. The Graphviz layout programs take descriptions of graphs in a simple text language, and make diagrams in useful formats, such as images and SVG for web pages; PDF or Postscript for inclusion in other documents; or display in an interactive graph browser.

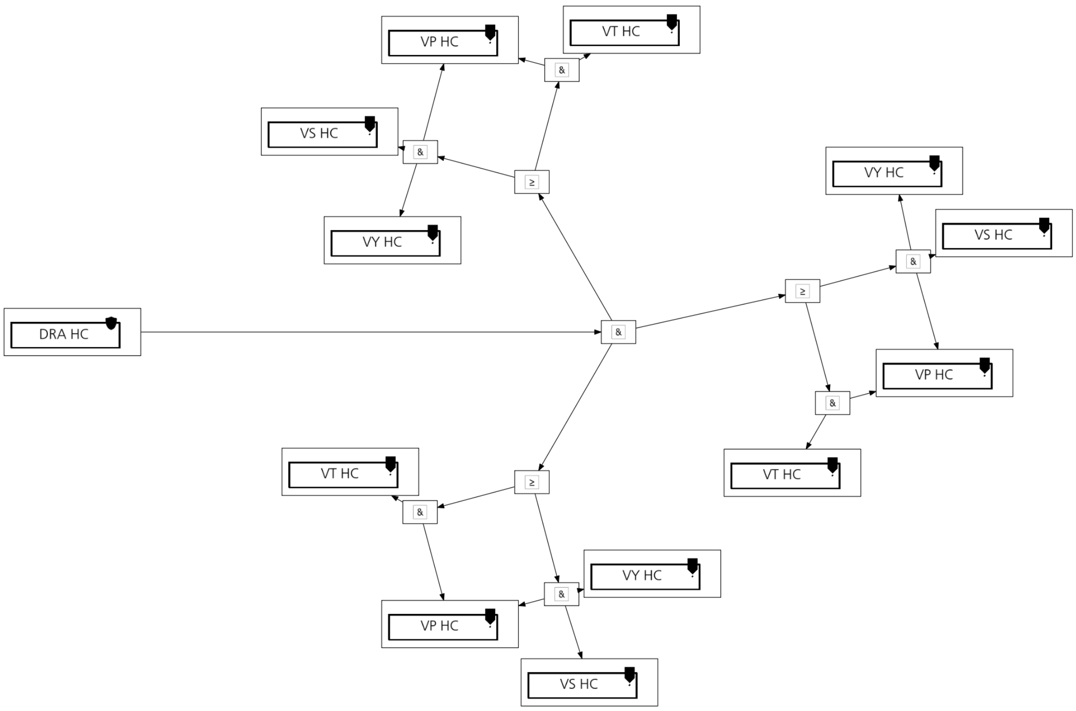

FIGURE 4.8: ConSert Tree for a high confidence Risk assessment with 3 vehicles

An engine and the format (PNG) for the tree list is selected. The image that is produced as an outcome is then rendered to the pygame visualization window in order to visualize the ConSert trees. Before connecting the ConSerts evaluation and visualization services to SUMO simulator, neutral ConSerts with one vehicle and three vehicles for different confidence levels are first visualized for better understanding.

The ConSert philosophy is that the tree is created during development time and the nodes in the tree are evaluated during runtime. Consider the ConSert tree in the figure 4.7. It represents a ConSert tree for a high confidence risk assessment with one vehicle. It can be clearly seen that in order to perform a high confidence Dynamic Risk Assessment i.e, for the DRA HC guarantee to be fulfilled, one of the "and" conditions has to become true. If VS HC, VY HC and VP HC demands are met, reachable area risk assessment for high confidence is possible. If VT HC and VP HC demands are met, intended trajectory risk assessment for high confidence is possible. The same is extended to three vehicles as in the figure 4.8. In this case with respect to demands for each vehicle atleast one of the two risk assessments i.e, either intended trajectory risk assessment or reachable area risk assessment should be possible. Only then the "and" condition gets satisfied to perform a DRA for high confidence.

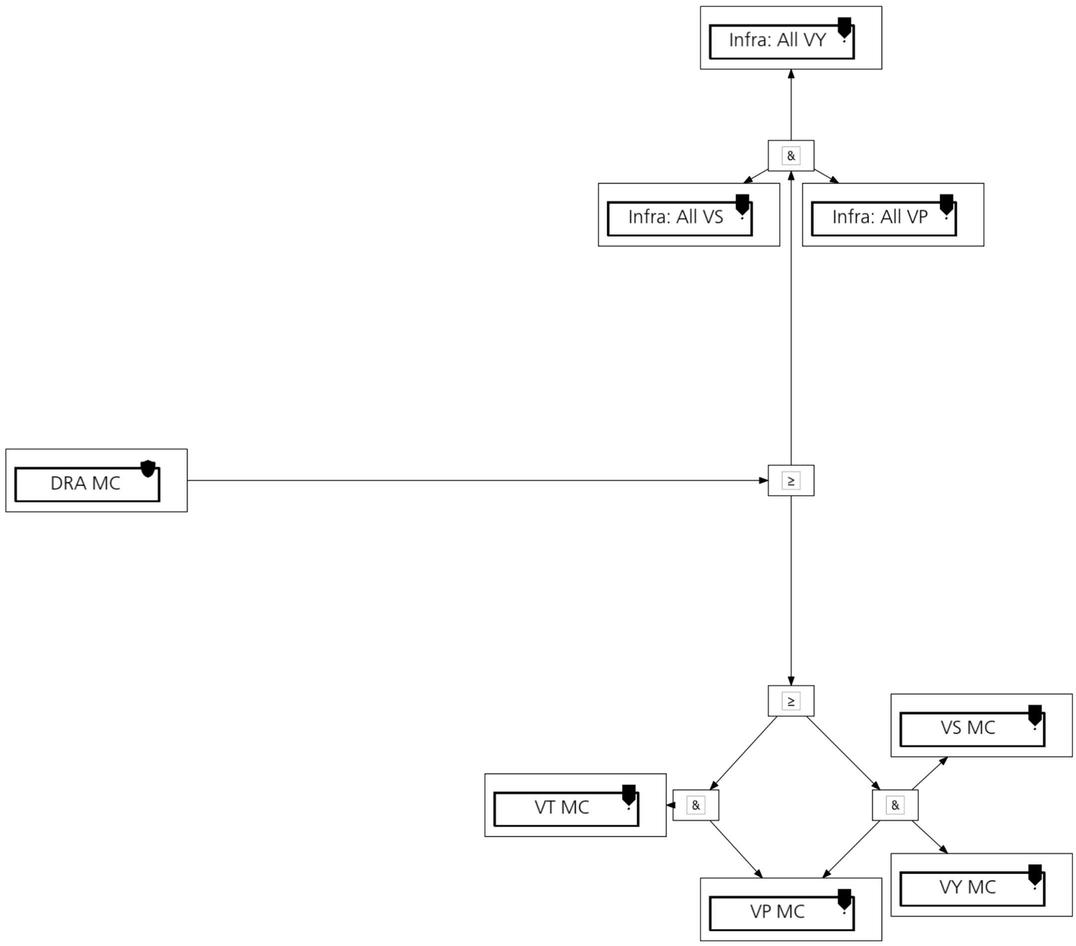

FIGURE 4.9: ConSert Tree for a mid confidence Risk assessment with 1 vehicle

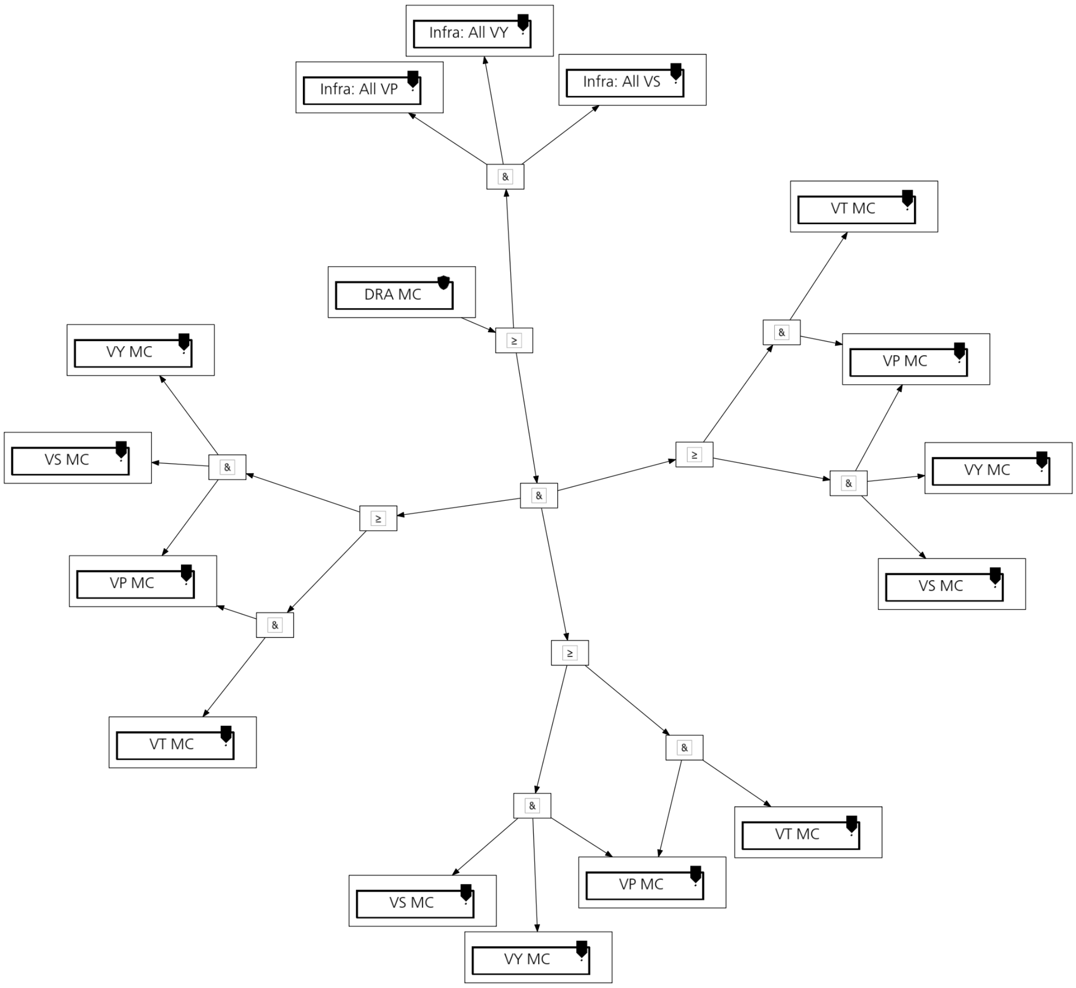

FIGURE 4.10: ConSert Tree for a mid confidence Risk assessment with 3 vehicles

Consider the ConSert tree in the figure 4.9. It represents a ConSert tree for a mid confidence risk assessment with one vehicle. Here, in this case, in order to perform a mid confidence Dynamic Risk Assessment i.e, for the DRA MC guarantee to be fulfilled, one of the conditions either side of "or" has to become true. If VS MC, VY MC and VP MC demands are met, reachable area risk assessment for mid confidence is possible. If VT MC and VP MC demands are met, intended trajectory risk assessment for mid confidence is possible. If both the conditions fail, there is still a chance to perform mid confidence risk assessment if the infrastructure is able to meet the demands Infra: All VS, All VY and All VP. The same is extended to three vehicles as in the figure 4.8. As the infrastructure is independent of the vehicles, it is only represented once for any number of vehicles.

Next, consider the ConSert tree in the figure 4.11. It represents a ConSert tree

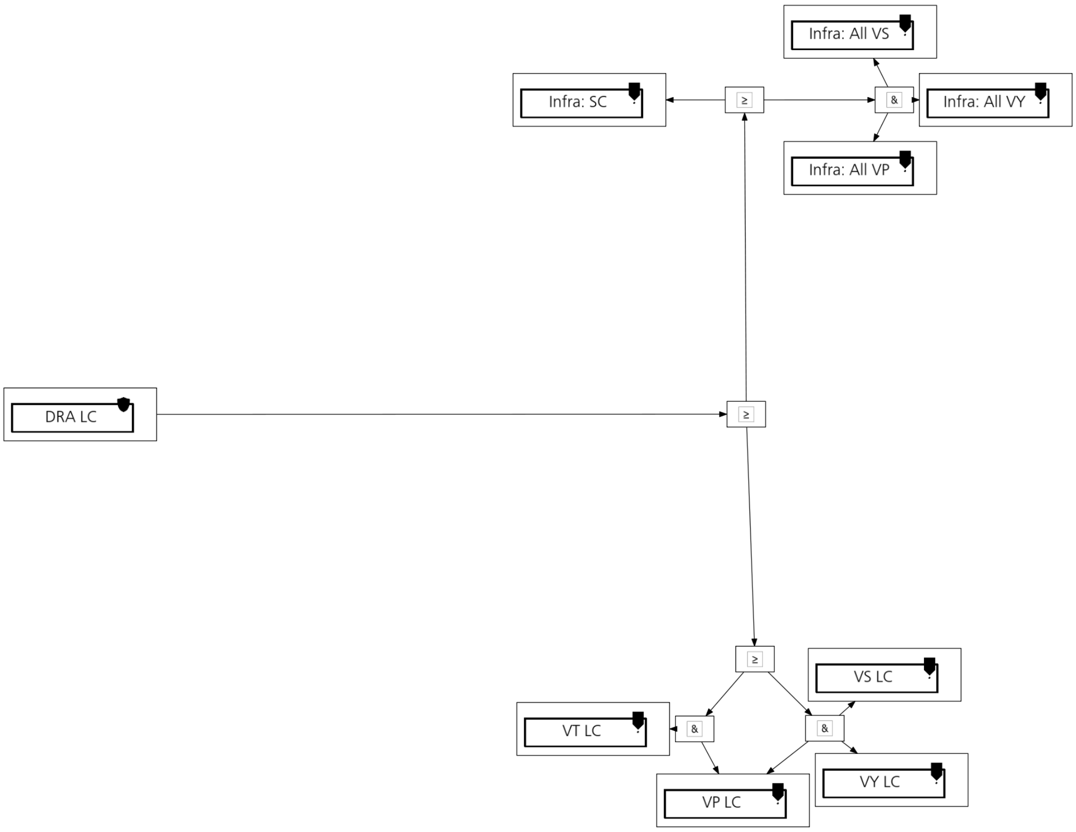

FIGURE 4.11: ConSert Tree for a low confidence Risk assessment with 1 vehicle

for a low confidence risk assessment with one vehicle. Also, in this case, in order to perform a mid confidence Dynamic Risk Assessment i.e, for the DRA MC guarantee to be fulfilled, one of the conditions either side of "or" has to become true. If VS LC, VY LC and VP LC demands are met, reachable area risk assessment for low confidence is possible. If VT LC and VP LC demands are met, intended trajectory risk assessment for low confidence is possible. If both the conditions fail, there is still a chance to perform low confidence risk assessment if the infrastructure is able to meet all the demands Infra: All VS, All VY and All VP or just meet the demand Infra: SC. Stereo camera is only used for low confidence risk assessment when no information is available at all. The same is extended to three vehicles as in the figure 4.12. As the infrastructure is independent of the vehicles, it is only represented once for any number of vehicles.

If none of the conditions are met, then a risk assessment cannot be performed.

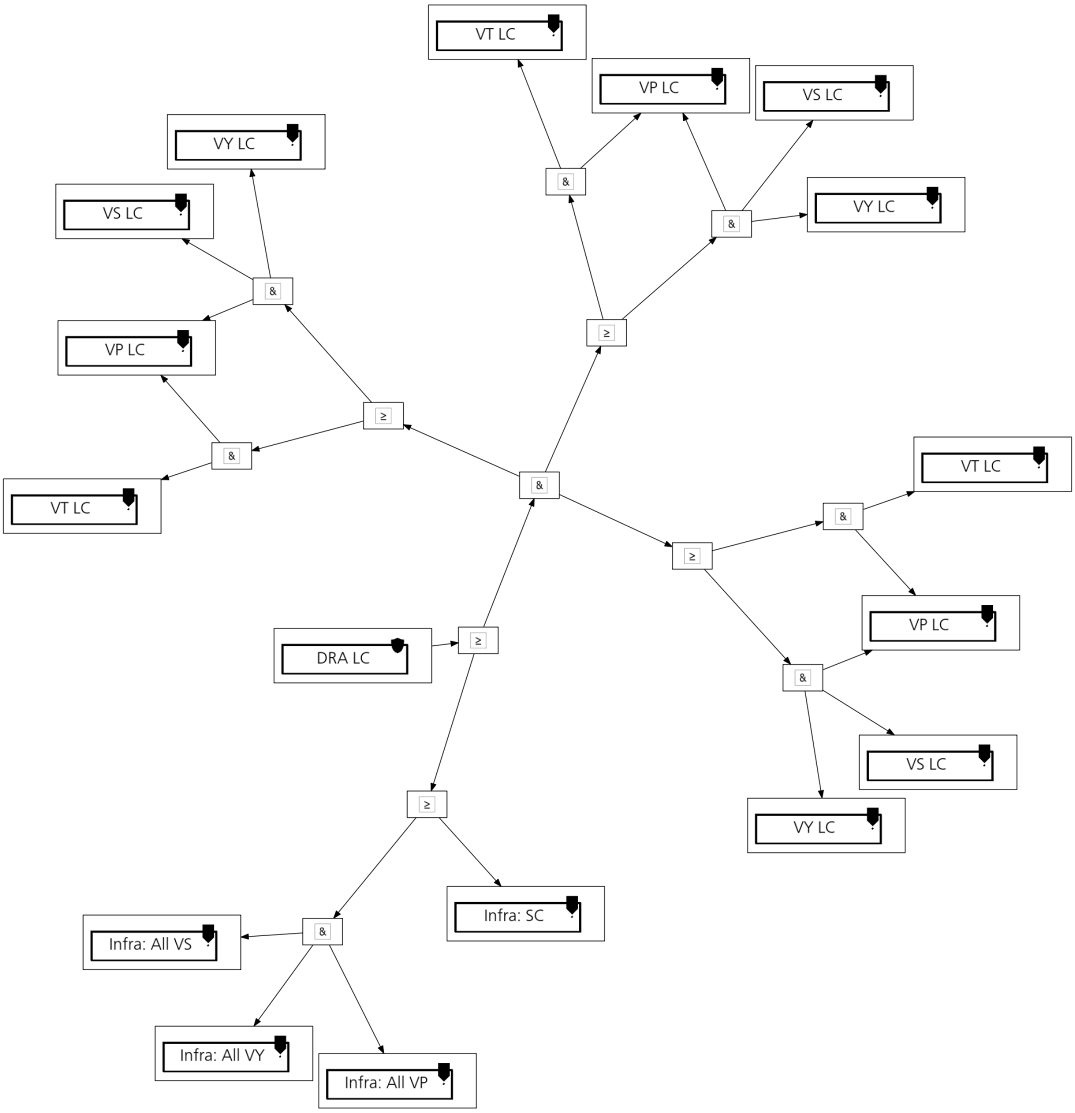

FIGURE 4.12: ConSert Tree for a low confidence Risk assessment with 3 vehicles

4.3 Mission level Dynamic Risk Assessment

The function calculate_general_ttc_online (current_occ_map, subject_vehicle_id, radius, confidence_level) takes current occupancy map, vehicle ID, radius (value considered is 50) and confidence level (0, 1 or 2) as input and is used for the calculation of TTC in the current work. Further calculation of the risk metrics GTTC, ITTC and MTTC are all based on this TTC value. The default value of TTC is set to 10000.

The result from the evaluation functions of the ConSerts Evaluate_DRA_Mid _Conf(vehID), Evaluate_DRA_Low_Conf(vehID) and Evaluate_DRA_High_Conf (vehID) are used to determine on which confidence level the risk metric calculation shall be performed. The function definition create_and_visualize_occ_map_with_metric(prediction_horizon) is defined for this purpose. The advantage of using ConSerts is that the performance of the DRA system can be adapted dynamically without considering the worst case for all scenarios. In regard to this, the results of the evaluation functions and how the ConSerts benefit us is discussed in more detail in chapter 5.

Chapter 5

Evaluation

This chapter explains the visualization of risk metric on the occupancy map in section 5.1 and the mission level ConSerts evaluation and visualization in the section 5.2. For a better understanding, results from the current work with examples are explained considering different scenarios.

5.1 Visualization of the risk metric on the occupancy map

To visualize the risk metric Mission level time to collision (MTTC) that is to be used for DRA at the intersection along with the AVG value that is calculated for estimating the criticality of the situation, a function called create_and_visualize_occ_map _with_metric(prediction_horizon) is defined. As discussed in the previous chapters, the prediction horizon value is considered as 1000. In this function, the ConSert evaluation functions for vehicle types are also defined. Evaluate_DRA_Low_Conf (vehID), Evaluate_DRA_Mid_Conf (vehID) and Evaluate_DRA_High_Conf (vehID) are the functions. The corresponding TTC's are calculated only if these functions return true else, default value is used for displaying the results on the Occupancy map. The code can be seen in the figures 5.1 and 5.2.

The occupancy map has to be updated for every time step and also the metic values have to be refreshed. For this reason, a function update_visualization_confidence (current_occ_map, current_metric_value_low_conf, current_metric_value_mid_conf, current_metric_value_high_conf, metric_type_low_conf, metric_type_mid_conf, metric_type_high_conf, confidence_level_to_use, average_low, average_mid, average

```python
def create_and_visualize_occ_map_with_metric(prediction_horizon):
    metrics_to_use = [0, 1, 2]
    ITTC_low = []
    ITTC_mid = []
    ITTC_high = []
    MTTC_avg = []
    MTTC_min = []

    current_occ_map = occ_map.create_online_occupancy_map_confidence(managed_vehicles, static_vehicle_information,
                                                                      dynamic_vehicle_information,
                                                                      prediction_horizon,
                                                                      traci.simulation.getCurrentTime())

    for vehID in managed_vehicles:
        if (Evaluate_DRA_Low_Conf(vehID)):
            ITTC_low.append(om.calculate_general_ttc_online(current_occ_map,vehID,50,0))
            #print(" GTTC_LOW: " ,GTTC_low)
        else:
            metrics_to_use[2] = -1

        if (Evaluate_DRA_Mid_Conf(vehID)):
            ITTC_mid.append(om.calculate_general_ttc_online(current_occ_map, vehID, 50, 1))
            #print(" GTTC_MID: ", GTTC_mid)
        else:
            metrics_to_use[2] = -1

        if (Evaluate_DRA_High_Conf(vehID)):
            ITTC_high.append(om.calculate_general_ttc_online(current_occ_map, vehID, 50, 2))
            #print(" GTTC_HIGH: ", GTTC_high)
        else:
            metrics_to_use[2] = -1
        print(vehID)
    array = [ITTC_low,ITTC_mid,ITTC_high]
    print("Array: ",array)
```

FIGURE 5.1: create_and_visualize_occ_map_with_metric ( ) function I

```python
if not(ITTC_low == [] and ITTC_mid == [] and ITTC_high == []):
    MTTC_min = [min(array[0]),min(array[1]),min(array[2])]
    print("MTTC_min: ", MTTC_min)
    Avg_high = np.mean(ITTC_high)
    Avg_mid = np.mean(ITTC_mid)
    Avg_low = np.mean(ITTC_low)
    MTTC_avg = [Avg_low,Avg_mid,Avg_high]
    vis_occ_map.update_visualization_confidence(current_occ_map, MTTC_min[0], MTTC_min[1], MTTC_min[2], "MTTC", "MTTC", "MTTC",
                                                metrics_to_use, MTTC_avg[0], MTTC_avg[1], MTTC_avg[2], 2)
else:
    vis_occ_map.update_visualization_confidence(current_occ_map, 10000, 10000, 10000, "MTTC",
                                                    "MTTC", "MTTC",
                                                    metrics_to_use, 10000, 10000, 10000, 2)
```

FIGURE 5.2: create_and_visualize_occ_map_with_metric ( ) function II

_high, driving situation) is defined to update the visualization. The input parameters are the current occupancy map, low confidence risk metric value, mid confidence risk metric value, high confidence risk metric value, type of risk metric used in case of low confidence (MTTC is used for all confidence levels), risk metric used in case of mid confidence, risk metric used in case of high confidence, AVG value for low confidence, AVG value for mid confidence, AVG value for high confidence and the driving situation which is arbitrary in our current scenario. The function can be seen in the figures 5.3 and 5.4.

The result is displayed in the form of (MTTC for the confidence level low/ mid/ high, corresponding AVG value for low/ mid/ high confidence levels)).

```
def update_visualization_confidence(current_occ_map, current_metric_value_low_conf, current_metric_value_mid_conf,
                                    current_metric_value_high_conf, metric_type_low_conf, metric_type_mid_conf,
                                    metric_type_high_conf, confidence_level_to_use, average_low,average_mid,average_high, driving_situation):
    game_display.fill(gray)
    average_high = int(average_high)
    average_low = int(average_low)
    average_mid = int(average_mid)


    current_metric_value_low_conf_round = decide_confidence_value(current_metric_value_low_conf, current_metric_value_mid_conf,
                                                                  current_metric_value_high_conf, 0, confidence_level_to_use[0])
    if str(current_metric_value_low_conf_round) == "nan":
        average_low = "nan"
        metric_low_conf_text = myfont.render(metric_type_low_conf + " (Low Confidence): "+ "("+
                                             str(current_metric_value_low_conf_round) + ", "+
                                             str(average_low)+")", False, (0, 0, 0))
        game_display.blit(metric_low_conf_text, (0, 0))
    else:
        metric_low_conf_text = myfont.render(
            metric_type_low_conf + " (Low Confidence): " + "(" + str(current_metric_value_low_conf_round) + ", " +
            str(average_low) + ")", False, (0, 0, 0))
        game_display.blit(metric_low_conf_text, (0, 0))

    current_metric_value_mid_conf_round = decide_confidence_value(current_metric_value_low_conf, current_metric_value_mid_conf,
                                                                  current_metric_value_high_conf, 1, confidence_level_to_use[1])
```

FIGURE 5.3: update_visualization_confidence ( ) function I

```
if str(current_metric_value_mid_conf_round) == "nan":
    average_mid = "nan"
    metric_low_conf_text = myfont.render(metric_type_mid_conf + " (Mid Confidence): "+
                                         "("+str(current_metric_value_mid_conf_round) + ", " +
                                         str(average_mid)+")", False, (0, 0, 0))
    game_display.blit(metric_low_conf_text, (0, 30))
else:
    metric_low_conf_text = myfont.render(
        metric_type_mid_conf + " (Mid Confidence): " + "(" + str(current_metric_value_mid_conf_round) + ", " +
        str(average_mid) + ")", False, (0, 0, 0))
    game_display.blit(metric_low_conf_text, (0, 30))

current_metric_value_high_conf_round = decide_confidence_value(current_metric_value_low_conf, current_metric_value_mid_conf,
                                                               current_metric_value_high_conf, 2, confidence_level_to_use[2])
if str(current_metric_value_high_conf_round) == "nan":
    average_high = "nan"
    metric_low_conf_text = myfont.render(metric_type_high_conf + " (High Confidence): "+
                                         "("+str(current_metric_value_high_conf_round)+ ", "+
                                         str(average_high) +")",  False, (0, 0, 0))
    game_display.blit(metric_low_conf_text, (0, 60))
else:
    metric_low_conf_text = myfont.render(
        metric_type_high_conf + " (High Confidence): " + "("+str(current_metric_value_high_conf_round) +
        ", " + str(average_high)+")", False, (0, 0, 0))
    game_display.blit(metric_low_conf_text, (0, 60))

driving_situation_label = driving_situation_to_string(driving_situation)

if not driving_situation_label == "":
    driving_situation_text = myfont.render("Situation: " + driving_situation_label, False, (0, 0, 0))
    game_display.blit(driving_situation_text, (650, 0))
```

FIGURE 5.4: update_visualization_confidence ( ) function II

5.2 Mission level ConSerts evaluation and visualization

The evaluation of ConSerts at runtime is based on the runtime models and dedicated mechanisms and protocols. These are partially dependent on the characteristics of the system such as system architecture, approach for self-adaptivity, etc (Schneider and Trapp, 2013a). It also depends the capabilities of the vehicles at the intersection. The evaluation results of ConSert trees is used to determine on which confidence level the risk metric i.e, MTTC calculation shall be performed.

As mentioned in the chapter 4, thrift is used to connect the ConSert RTE visualization and evaluation services with the RMC. A separate client called kalyani _client.py is defined to call the evaluation client and the visualization client of the ConSerts and also to make a connection to RMC, which makes a connection to the cross.sumocfg file that opens up the simulation screen window of SUMO simulator.

```python
def Evaluate_DRA_Low_Conf(self, vehicle_types):
    #for now the existance of the infrastructure services always allow a low confidence risk assessment

    return True

def Evaluate_DRA_Mid_Conf(self, vehicle_types):
    #for now the existance of the infrastructure services always allow a mid confidence risk assessment

    return True

def Evaluate_DRA_High_Conf(self, vehicle_types):
    #every vehicle needs to be of either type RA_HC or All_HC to enable the high conf DRA
    if "IT_MC" in vehicle_types or "Legacy" in vehicle_types:
        return False
    else:
        return True
```

FIGURE 5.5: ConSerts RTE evaluation functions for low, mid and high confidence levels

Evaluations are performed for high, mid and low confidences for the vehicle types at the intersection and corresponding visualization result is seen on the pygame display window. If the ConSerts evaluation engine returns the result that the DRA for high, mid or low confidence can be performed then the corresponding risk metric MTTC value and AVG value are seen on the occupancy map. If the ConSerts evaluation engine returns the result that the DRA cannot be performed for the current time step, the corrresponding MTTC and AVG values are both displayed as "nan" on the occupancy map for that particular confidence level.

The functions corresponding to the evaluation of DRA for low, mid and high confidence levels can be seen in the figure 5.5. It is observed that in Evaluate_DRA_High _Conf(vehicle_types) function, if there is a vehicle either of type IT_MC or Legacy, because of the information from the infrastructure, it returns false.

For a better understanding and differentiation, few timed scenarios are taken as an example to explain the results of the work. The ConSert trees with high confidence are displayed as it is difficult to display compactly for all the three confidence levles. Considering the scenario in the figure 5.6, there are three vehicles at the intersection one of the type RA_HC in pink color and two of the type All_HC in blue color. As the RA_HC can only perform reachable area risk assessment, the kamm's circle is expanded to the given radius. It can be clearly seen that there are three different confidence level circles. The outer most corresponds to high confidence, the inner most to low confidence and the middle one corresponds to mid confidence.

It is noticed that there is an overlap between the future trajectory of the All_HC

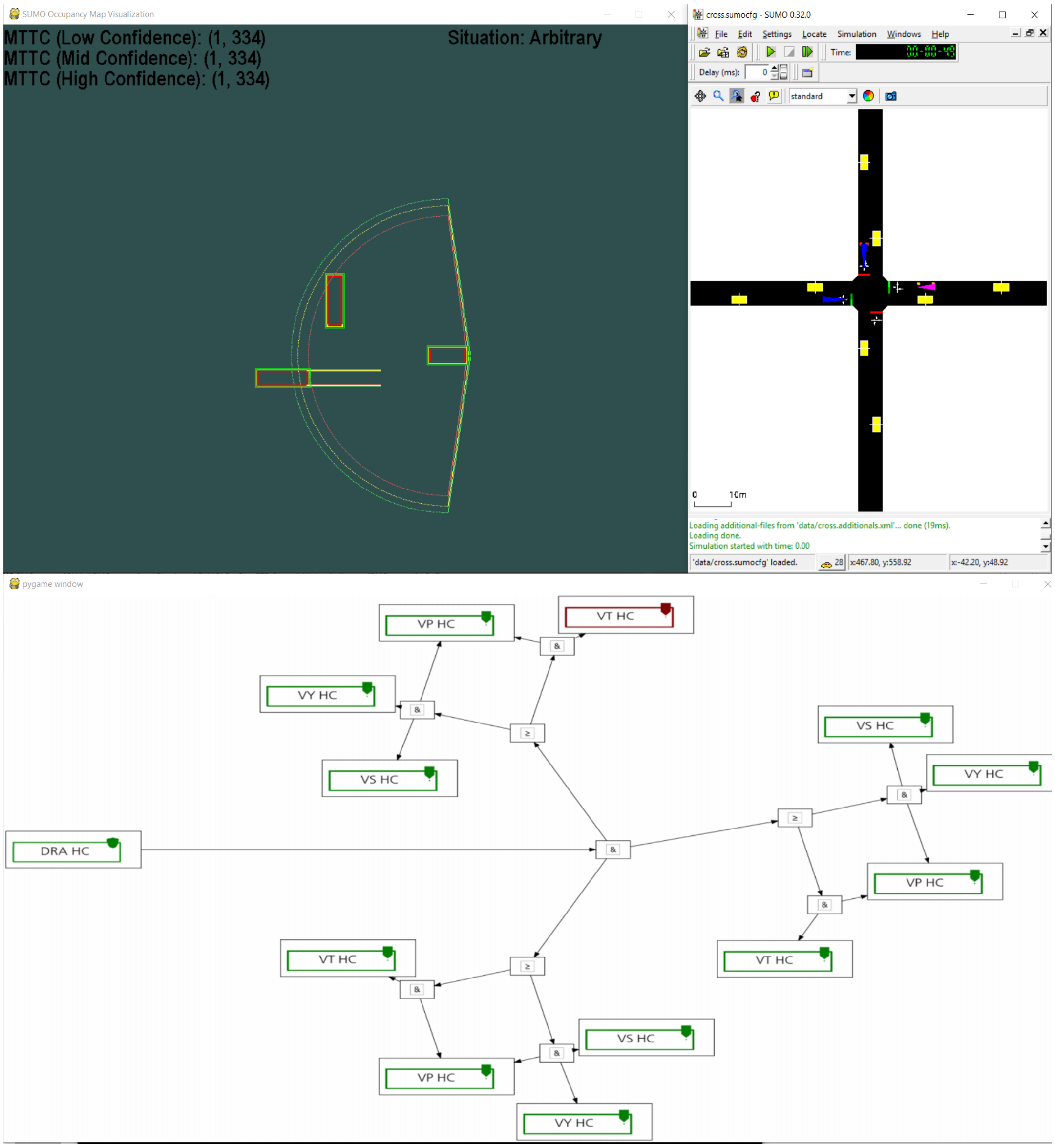

FIGURE 5.6: DRA being performed with the vehicle types RA_HC and All_HC at the intersection

vehicle type and the Kamm's circle of RA_HC vehicle type. The ConSert tree corresponding to that time step for the vehicle types at the intersection returns that the DRA for high confidence can be performed. This can easily be noticed with the green color guarantee DRA HC being fulfilled. Red color implies that the demand or the guarantee is not fulfilled. Therefore, it now means that even mid and low risk DRA can be performed.

The trajectories created are segmented and the first segment (which is intersecting here) has because of the worst-case assumptions a prediction value of 1 ms. In theory it should be 0 ms, but the 0 ms prediction time is reserved for the current cells. Thus, a decision has been made to use 1 ms as the TTC value whenever there is overlapping. The reason is that for the points nearest to the predicted trajectory of the car 1000ms would be wrong (e.g. the car doesn't need 1000ms to reach a point 10cm in front of it). Hence 1 ms is used as the worst case TTC value.

The corresponding risk metric values and the AVG values that are calculated are updated on the occupancy map. The values are same for all the three confidence levels in this scenario case.

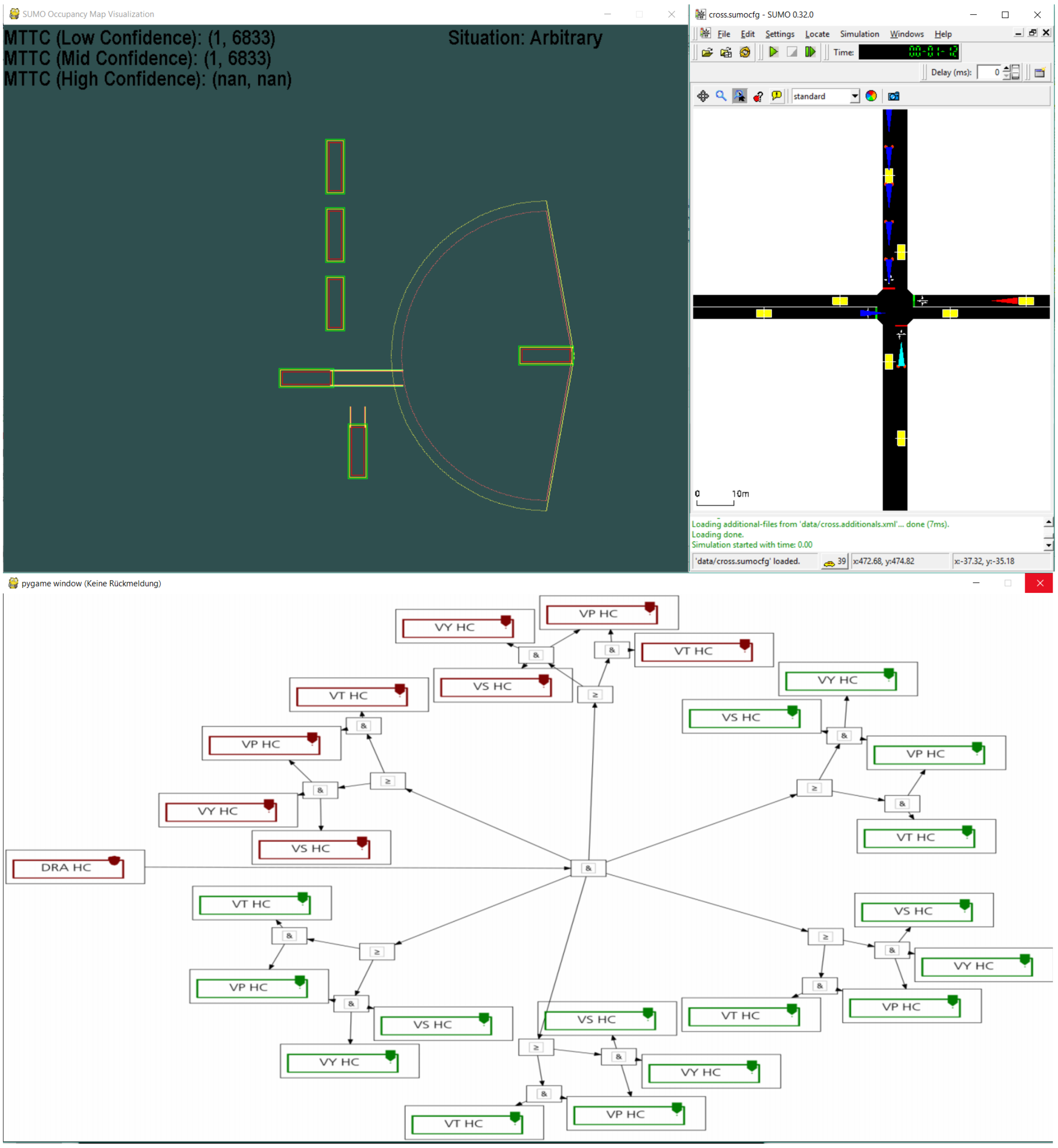

FIGURE 5.7: DRA being performed with the vehivle types All_HC, IT_MC and Legacy at the intersection

Considering the scenario in the figure 5.7, it is seen that vehicles of the vehicle types All_HC, IT_MC and Legacy are at the intersection. From the ConSerts display window, it is observed that the DRA HC guarantee box is red in color. Which implies that DRA for high confidence cannot be performed at the intersection. Hence, it is observed that the MTTC and AVG value for high confidence, are displayed as "nan".

Also in the occupancy map, it can be observed that the vehicle type Legacy does not have the high confidence Kamm's circle. Same goes with the predicted trajectories of IT_MC type. Since there is an over lap of the future trajectory of the vehicle type IT_MC with legacy, the corresponding MTTC and AVG values for mid and low confidence levels are displayed on the occupancy map.

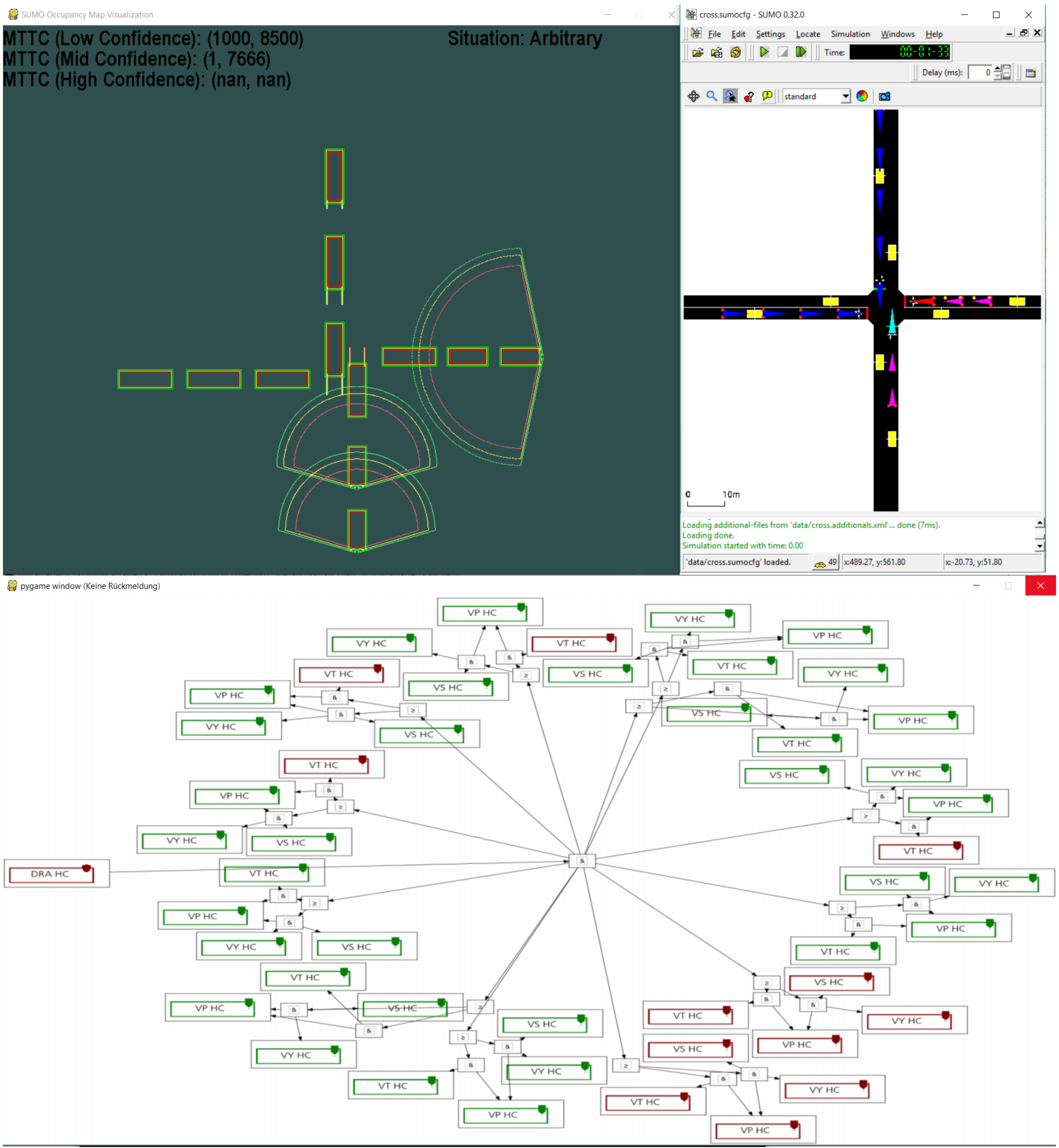

FIGURE 5.8: DRA being performed with the vehicle types All_HC, IT_MC, RA_HC and Legacy at the intersection

Lastly, considering the scenario in the figure 5.8, it is seen that vehicles of all the types All_HC, IT_MC, RA_HC and Legacy are at the intersection for that particular time step. The ConSerts evaluation has returned that the DRA for high confidence cannot be performed which is clearly understood because of the red box of DRA HC in the ConSert visualization window. Hence, correspondingly, the MTTC and AVG values for high confidence are displayed as "nan".

If closely observed, first pink vehicle following the SN route is of the type RA_HC and it is overlapping with first blue vehicle following the route NS which is of the type All_HC.Hence, MTTC value for mid confidence is i ms and, the corresponding MTTC and AVG values are displayed on the occupancy map.

If only the two pink vehicles following the route SN, both of the type RA_HC are considered, there is an overlap of the Kamm's circles of both the vehicles. There is no other overlap in the occupancy map. Hence, the MTTC value for low confidence is 1000 ms. The corresponding MTTC and AVG values are displayed on the occupancy map.

Chapter 6

Conclusion and Future work

6.1 Conclusion

In this work, an approach for calculating the global/ mission level risk for the considered intersection with multiple vehicles at a current time step has been proposed. Amongst the various existing automotive collision risk metrics, a general version of TTC has been taken as the base metric, regards to this, a new risk metric MTTC (Mission level Time To Collision) has been defined. In order to differentiate the criticality of the situations and scenarios, an extra parameter AVG has been introduced. MTTC remaining same, lower the AVG value, higher is the criticality.

Conditional Safety Certificates (ConSerts) have been visualized and ConSerts evaluation is used to realize if Dynamic Risk Assessment can be done on high, mid and low confidence levels at runtime. In other words, ConSerts help determine the confidence level and the method of the DRA based on the capabilities of the vehicles at the intersection.

Based on the architecture represented by a ConSert tree in the figure 3.5, for the proposed approach, different vehicle types have been defined with certain demands and guarantees met from the vehicle perspective. Either an intended trajectory prediction or reachable area prediction based risk assessments are used for the different vehicle types. The driving situation is considered arbitrary and the whole intersection scenario is modelled in SUMO simulator.

For the future trajectories in case of intended trajectory prediction method, the dump file that is generated by SUMO simulator when the simulation is run is used. Whereas, for the reachable area prediction method, Kamm's circle which is further parameterized is used.

The whole scenario from the SUMO simulator is imported into the RMC. The information is translated to a Grid map in which the world is discretized in fixed cells. For each of those cells, occupancy information is calculated and the result is visualized on the occupancy map. The figures 5.6, 5.7 and 5.8 compactly visualize the whole work. The current work thus helps to understand how a Dynamic Risk Assessment can be carried out at an intersection using SUMO simulator, with the help of concepts related to ConSerts and Risk Metric Calculator for calculation of risk metrics.

6.2 Future work

There are few limitations to the work. While calculating the MTTC, the value 1000 ms was used for Kamm's circle and 1 ms for predicted trajectories. This is because only the border points are considered for computational efficiency. Also, for instance it is known that a vehicle moving at a given speed will need a certain time to fully stop and that the curvature of its trajectory has to be under a certain value in order to keep stability but this is not clearly addressed in the current work. The intersection on which the Dynamic Risk Assessment (DRA) has been performed is a simple one. There could be many variables if a complex intersection is chosen such as pedestrians, cyclists, traffic signs, etc,. There could be other collision risk metrics which have to be studied in order to perform a dynamic risk assessment.

Most of the work done on automotive collision metrics always considered a simple intersection. Also very little work has been done for calculation of a global risk metric for an intersection level. There is a scope to think of further ways to arrive at a global level risk metric calculation better than the Mission Level Time To Collision (MTTC) calculated in the current work. Furthermore, the concept of ConSerts for evaluating DRA at runtime provides a promising means to ensure safety in open adaptive systems. Thus, understanding the functionality and integrating the concepts for complex scenarios could be considered as a part of future work.

Bibliography

Transportation Engineering I (Dec. 2009). NPTEL. Chapter 39 Traffic intersections. URL: https://nptel.ac.in/courses/105101087/40 (visited on 11/06/2018).

Chen, L. and C. Englund (2016). “Cooperative Intersection Management: A Survey”. In: *IEEE Transactions on Intelligent Transportation Systems* 17.2, pp. 570–586. ISSN: 1524-9050. DOI: 10.1109/TITS.2015.2471812.

Krajzewicz, Daniel, Jakob Erdmann, Michael Behrisch, and Laura Bieker (2012). “Recent Development and Applications of SUMO - Simulation of Urban MObility”. In: *International Journal On Advances in Systems and Measurements*. International Journal On Advances in Systems and Measurements 5.3&4, pp. 128–138. URL: http://elib.dlr.de/80483/;https://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/ (visited on 11/06/2018).

Zheng, Yi, Lisheng Jin, Yuying Jiang, Fangrong Wang, Xin Guan, Shengyuan Ji, and Jun Xu (2017). “Research on cooperative vehicle intersection control scheme without using traffic lights under the connected vehicles environment”. In: *Advances in Mechanical Engineering* 9.8, p. 1687814017719219.

Li, Bryon (June 2000). URL: http://fresno.ts.odu.edu/newitsd/ITS_Serv_Tech/collision_avoidance/coll_avoid_inter/collision_avoidance_intersection_summary.html (visited on 11/06/2018).

Huang, Rulin, Huawei Liang, Pan Zhao, Biao Yu, and Xinli Geng (2017). “Intent-estimation-and motion-model-based collision avoidance method for autonomous vehicles in urban environments”. In: *Applied Sciences* 7.5, p. 457.

Hafner, Michael R, Drew Cunningham, Lorenzo Caminiti, and Domitilla Del Vecchio (2013). “Cooperative Collision Avoidance at Intersections: Algorithms and Experiments.” In: *IEEE trans. intelligent transportation systems* 14.3, pp. 1162–1175.

Feth, Patrik, Daniel Schneider, and Rasmus Adler (2017). “A Conceptual Safety Supervisor Definition and Evaluation Framework for Autonomous Systems”. In:

Computer Safety, Reliability, and Security: 36th International Conference, SAFECOMP 2017, Trento, Italy, September 13-15, 2017, Proceedings. Ed. by Stefano Tonetta, Erwin Schoitsch, and Friedemann Bitsch. Cham: Springer International Publishing, pp. 135–148. DOI: 10.1007/978-3-319-66266-4_9. URL: https://doi.org/10.1007/978-3-319-66266-4_9.

Lefèvre, Stéphanie, Dizan Vasquez, and Christian Laugier (2014). "A survey on motion prediction and risk assessment for intelligent vehicles". In: *Robomech Journal* 1.1, p. 1.

Transport, European Conference of Ministers of (2004). *Assessment and decision making for sustainable transport*. Organization for Economic.

Jansson, Jonas (2005). "Collision Avoidance Theory: With application to automotive collision mitigation". PhD thesis. Linköping University Electronic Press.

Katrakazas, Christos, Mohammed Quddus, Wen-Hua Chen, and Lipika Deka (2015). "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions". In: *Transportation Research Part C: Emerging Technologies* 60, pp. 416 –442. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2015.09.011. URL: http://www.sciencedirect.com/science/article/pii/S0968090X15003447.

Schreier Matthias Willert, Volker Adamy Jurgen (2016). "An Integrated Approach to Maneuver-Based Trajectory Prediction and Criticality Assessment in Arbitrary Road Environments." In: *IEEE Transactions on Intelligent Transportation System*. DOI: 17.2751-2766.10.1109/TITS.2016.2522507..

Hayward, John C (1972). "Near miss determination through use of a scale of danger". In:

Hydén, Christer (1987). "The development of a method for traffic safety evaluation: The Swedish Traffic Conflicts Technique". In: *Bulletin Lund Institute of Technology, Department* 70.

Kruysse, Herman W (1991). "The subjective evaluation of traffic conflicts based on an internal concept of dangerousness". In: *Accident Analysis & Prevention* 23.1, pp. 53–65.

Tiwari, Geetam, Dinesh Mohan, and Joseph Fazio (1998). "Conflict analysis for prediction of fatal crash locations in mixed traffic streams". In:

Hou, Jia, George List, and Xiucheng Guo (Dec. 2014). "New Algorithms for Computing the Time-to-Collision in Freeway Traffic Simulation Models". In: *Computational intelligence and neuroscience* 2014, p. 761047. DOI: 10.1155/2014/761047.

Berthelot, Adam, Andreas Tamke, Thao Dang, and Gabi Breuel (2012). "A novel approach for the probabilistic computation of time-to-collision". In: *Intelligent Vehicles Symposium (IV), 2012 IEEE*. IEEE, pp. 1173–1178.

Hillenbrand, Jrg, Andreas M Spieker, and Kristian Kroschel (2006). "A multilevel collision mitigation approach—Its situation assessment, decision making, and performance tradeoffs". In: *IEEE Transactions on intelligent transportation systems* 7.4, pp. 528–540.

Horst, RVD (1991). "Time-to-collision as a cue for decision-making in braking". In: *Vision in Vehicles–III*.

Kaempchen, Nico, Bruno Schiele, and Klaus Dietmayer (2009). "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios". In: *IEEE Transactions on Intelligent Transportation Systems* 10.4, pp. 678–687.

Ferguson, Dave, Michael Darms, Chris Urmson, and Sascha Kolski (2008). "Detection, prediction, and avoidance of dynamic obstacles in urban environments". In: *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, pp. 1149–1154.

Tamke, Andreas, Thao Dang, and Gabi Breuel (2011). "A flexible method for criticality assessment in driver assistance systems". In: *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, pp. 697–702.

Wachenfeld, Walther, Philipp Junietz, Raphael Wenzel, and Hermann Winner (2016). "The worst-time-to-collision metric for situation identification". In: *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, pp. 729–734.

Schramm, Dieter, Manfred Hiller, and Roberto Bardini (2014). "Vehicle dynamics". In: *Modeling and Simulation. Berlin, Heidelberg*, p. 151.

Schreier Matthias & Willert, Volker & Adamy Jürgen. (2014). "Bayesian, Maneuver-Based, Long-Term Trajectory Prediction and Criticality Assessment for Driver Assistance Systems." In: *17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014.* DOI: 10.1109/ITSC.2014.6957713..

Dijkstra, Atze and Hans Drolenga (2008). *Safety effects of route choice in a road network: Simulation of changing route choice*. Tech. rep. SWOV Institute for Road Safety Research.

Chen, Rong, Rini Sherony, and Hampton C. Gabler (Apr. 2016). "Comparison of Time to Collision and Enhanced Time to Collision at Brake Application during Normal Driving". In: *SAE Technical Paper*. SAE International. DOI: 10.4271/2016-01-1448. URL: https://doi.org/10.4271/2016-01-1448.

Vogel, Katja (2003). "A comparison of headway and time to collision as safety indicators". In: *Accident analysis & prevention* 35.3, pp. 427–433.

Jiménez, Felipe, José Eugenio Naranjo, and Fernando García (2013). "An improved method to calculate the time-to-collision of two vehicles". In: *International Journal of Intelligent Transportation Systems Research* 11.1, pp. 34–42.

Schneider, Daniel and Mario Trapp (2013b). "Engineering Conditional Safety Certificates for Open Adaptive Systems". In: *IFAC Proceedings Volumes* 46.22, pp. 139–144.

— (2013a). "Conditional safety certification of open adaptive systems". In: *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 8.2, p. 8.

Mekki-Mokhtar, Amina, Jean-Paul Blanquart, Jérémie Guiochet, David Powell, and Matthieu Roy (2012). "Safety trigger conditions for critical autonomous systems". In: *Dependable Computing (PRDC), 2012 IEEE 18th Pacific Rim International Symposium on*. IEEE, pp. 61–69.

Nunen, Ellen van, Dimitrios Tzempetzis, Gerald Koudijs, Henk Nijmeijer, and Mark van den Brand (2016). "Towards a safety mechanism for platooning". In: *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, pp. 502–507.